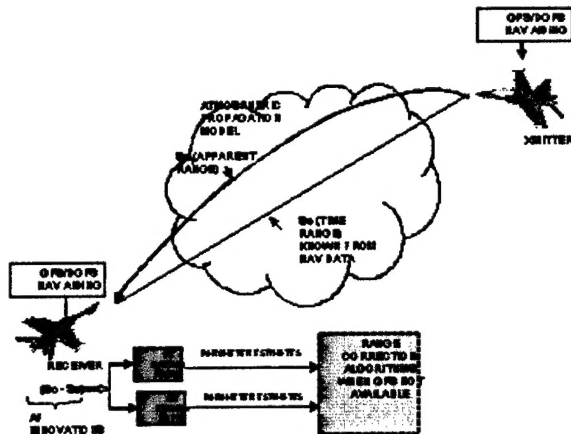BAE SYSTEMS

FINAL REPORT

IMPROVEMENT OF LINK 16 NAVIGATION VIA REAL-TIME ATMOSPHERIC MODELING

Joel D. Reiss
Ralph R. DeMarco

BAE SYSTEMS CNIR DIVISION
Wayne, NJ
30 September, 2003

Prepared for:
Program Officer
Office of Naval Research
Ballston Tower One
800 North Quincey Street
Arlington, VA 22217
Contract No. N00014-02-C-0418 CLIN2 CDRL0005

Attn: Dr. John Kim, Code 313, Telephone: (703) 696-4214

20040113 139

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.		
1. AGENCY USE ONLY *(Leave Blank)*	**2. REPORT DATE** 30 Sept. , 2003	**3. REPORT TYPE AND DATES COVERED** Scientific
4. TITLE AND SUBTITLE Improvement Of Link 16 Navigation Via Real-Time Atmospheric Modeling		**5. FUNDING NUMBERS** Contract No. SC133525-04 CLIN1
6. AUTHOR(S) J. Reiss and R. DeMarco		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSE(S) BAE Systems CNIR Division 164 Totowa Road Wayne, NJ 07474		**8. PERFORMING ORGANIZATION REPORT NUMBERS**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESSE(S) Dept. of the Navy Office of Naval Research 800 N. Quincy Street, Room 704 Arlington, VA 22217 Contract Manager: Dr. J. Kim		**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES		
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for Public Release – Distribution Unlimited		**12b. DISTRIBUTION CODE**

13. ABSTRACT *(Maximum 200 words)*

This study examines an algorithm, called the Atmospheric Filter, which improves upon the current method employed in the Link-16 MIDS terminal to compensate for the effects of atmospheric refraction on the range estimation to a Link-16 emitter. It is implemented in the form of a Kalman Filter to estimate the parameters of the refractivity modeled as an exponentially decaying function of altitude above sea level. The refractivity is used to correct the time a signal takes to travel from the emitter to the receiver. This then is used to estimate the range to the emitter. A truth model is used in this study to create realistic pseudo-measurements of the true range in order to determine the range errors due to the AF refractivity corrections. The goal is to reduce these errors by an order of magnitude and thus improve the performance of the existing navigation algorithms.

It is shown in this report that this simple exponential model agrees well with the well known, much more elaborate models of tropospheric refractivity. These models require large amounts of detailed meteorological data, using up considerable computing resources, both in data storage and also processing.

14. SUBJECT TERMS			**15. NUMBER OF PAGES** 258
			16. PRICE CODE
17. SECURITY CLASSIFICATION UNCLASSIFIED	**18. SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED	**19. SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED	**20. LIMITATION OF ABSTRACT** SAR
NSN 7450-01-280-5500			Standard Form 298 (Rev. 2-89) Prescribed by ANSI/NISO Std. Z39.18 298-102

1 TABLE OF CONTENTS

2 LIST OF FIGURES AND TABLES

3 ACKNOWLEDGMENTS

The work discussed herein is based on analyses originally performed by the late James Coleman of BAE Systems, including the development of the atmospheric models.

4 SUMMARY

This report describes an algorithm, called the Atmospheric Filter (AF), which is designed to improve upon the current method employed to compensate for the effects of atmospheric refraction on the range estimation to a Link-16 emitter.

Range measurement errors included in the Link-16 Precise Participant Location and Identification (PPLI) messages are the largest contributors to Link-16 navigation errors. The predominate contributor to the Link-16 range error is inadequately compensated signal refraction in the troposphere. This error affects the range estimates obtained from Time of Arrival (TOA) measurements in the PPLI message coming from a platform whose distance from the receiver it is desired to know more accurately.

Although Link-16 terminals currently include an approximate compensation for atmospheric refraction, this is a static, two-parameter model dating to 1971 which only involves average quantities.

The AF is implemented in the form of a Kalman Filter to estimate the parameters of the refractivity as modeled as an exponentially decaying function of altitude above sea level. The refractivity is used to correct the time a signal takes to travel from the emitter to the receiver. This then is used to estimate the range to the emitter. A truth model is used to create realistic pseudo-measurements of the true range in order to determine the range errors due to the AF refractivity corrections.

The goal of the algorithm is to reduce these errors by an order of magnitude. In the process of doing this, candidate atmospheric filter algorithms are defined, evaluated and compared for potential performance improvements.

This report describes the results of this study: the models of the atmosphere used in the AF algorithm, the refractivity models investigated, the Extended Kalman Filter implementation of the AF and sample results. Finally, conclusions and recommendations for further work are given.

It is shown in this report that this simple exponential model agrees with two well known, elaborate models of tropospheric refractivity. These other models, however, require *detailed meteorological data* which is nearly impossible to obtain on a real time basis operationally and which data would, even if available, use up considerable computing resources, both in data storage and also processing. *The exponential model does not require meteorological data; this is not needed as the signals themselves are refracted by the atmosphere, therefore they inherently provide the information needed to estimate*

refractivity. As the exponential model adequately describes the atmospheric refractivity, a Kalman filter can estimate the two constants of this model using as input signals from one or more Link 16 units. All that is needed is that the refractivity model be realistic; nothing more.

The Atmospheric Filter itself is shown to give good results. Its Kalman filter converges rapidly to good solutions in the scenarios it is expected to work in. It is suited to work best with emitters which are not at high angles to the receiving processor, due to the angular approximations made in developing the algorithm. Such angular relative geometries are usually the case, operationally, achieving the stated goals of the system.

5 INTRODUCTION

This report is concerned with the compensation of Link-16 "time of arrival" (TOA) measurements for the refractive effects of the atmosphere. Principally, the effect is to slightly reduce the speed of propagation of the radio frequency energy from is value in a vacuum. There is also a curvature effect that becomes significant at long ranges, due to the gradual change in the index of refraction, as will be noted elsewhere in this report.

Since Link-16 terminal time bases are synchronized to very high accuracy, the arrival time of a received message can be used by the terminal to measure the propagation time of the message. The terminal's navigation and passive synchronization functions process these TOA measurements, and by compensating for speed of light variations, convert them into range measurements. If Link-16 terminals operated in a vacuum, computation of range would be trivial, involving nothing more than the multiplication of the TOA measurement by the speed of light *in vacuo*, c. However, due to the atmosphere, the message travels at a speed c_{air}, slightly less than c, and, in addition, also follows a slightly curved path because the refractive properties of the air gradually change along the signal path. Compensation for these effects is mandatory, since they may cause large range errors under some conditions if not corrected.

The index of refraction, n, plays a major part in the development of atmospheric models, since c_{air} is given by c/n, where c is the speed of light in vacuo, equaling 299,792.458 meters/second. The index of refraction has a mean value of approximately 1.00003 at sea level, and typically varies over a range of 1.0002 to 1.0005 in the troposphere. Inaccurate estimates of n can cause sizeable errors in range estimates. Consider, for example, using an estimate of 1.00003 when the true index of refraction is 1.0002. At a range of 10 km, this mismodeling would cause a range error of only about 1 meter. At a range of 100 km, the corresponding range error grows to 10 meters. At longer ranges of 300 to 500 km, this error increases to 30 to 100 meters. These errors are unacceptable when applied to Link-16 navigation processing.

All existing Link-16 terminal assets utilize a simple two-parameter exponential (with altitude) compensation model for the index of refraction, which was first developed in 1971 and has not changed since that time. The primary disadvantage of this model is that the two static parameters that it uses reflect "one size fits all" constants averaged over standard

atmospheric conditions. While this model works reasonably well at shorter ranges, its use at ranges above 100 km can result in range errors exceeding 100 meters. Of course, it works best when the atmospheric conditions are average, but this does not obtain often.

The use of static parameters is not the only disadvantage of this approach. Even should the existing model provide a good estimate of the model parameters, it is fundamentally based upon a "flat earth" model, which becomes steadily more inaccurate as the ranges between transmitter and receiver increase. Differences between the true and estimated parameters will make range estimation performance even worse. Clearly, steps must be taken to improve both the estimate of the atmospheric parameters on which the index of refraction depends, as well as a replacement of the rigid "flat earth" model in favor of one which more closely resembles the dynamics of ray tracing (Snell's Law) which governs all radio propagation. In other words, better modeling is needed of (a) the atmosphere and (b) the signal propagation dynamics themselves.

In the flat earth (FE) atmospheric model, the earth's surface is flat and so are the layers of the atmosphere. In each layer, the index of refraction is taken to be constant. In the more realistic Round Earth (RE) or Ray model, the surface of the earth is spherical and the atmosphere consists of concentric spherical layers. Again, each of these layers is thin enough so that the index of refraction can be taken to be constant in it. Of course, the earth is not exactly spherical, but pear-shaped with some additionally varying curvature, but the RE model could feasibly be modified for deviations from a spherical shape, this is a subject for future analysis.

The atmospheric index of refraction is generally calculated using elaborate measurements of ambient meteorological data, such as pressure, temperature, and humidity. The index of refraction, n, varies by location on the earth, by season, and even by time of day. What are required are real time, efficient algorithms capable of accurately estimating key parametric models of refractivity without the need for hard to obtain atmospheric data. BAE SYSTEMS has developed such algorithms, based upon its experience with Sensor Registration. Sensor Registration has previously been used to calibrate sensors, such as long range radars, by measuring the range to a cooperating platform whose position is well known. Realizing that a Link-16 terminal measuring range is just another sensor, we apply the same principle to develop an Atmospheric Filter (AF) whose outputs are the desired atmospheric parameters.

A number of accurate models of tropospheric refractivity have been developed in the last forty years. Three of these models[1, 2, 3] will be examined here, see below. They are all shown to give equivalently good results. One is much better to implement operationally as it is simply a function of altitude and so is chosen to be so used. The other models require

---

[1] Collins, P. and Langley, R. "Limiting Factors in Tropospheric Propagation Delay Error Modeling for GPS Airborne Navigation," Presented at the Institute if Navigation 52nd Annual Meeting, Cambridge, MA, June 19-21, 1996.

[2] Flores, A., Ruffini, G. and Rius, A. "4D Tropospheric Tomography using GPS Estimated Slant Delays." Annales Geophysicae (2000) 18: 223 – 234.

[3] Thayer, D., An Improved Equation for the Radio Refractive Index of Air, Radio Sci., 9, 803-807, 1974.

input of detailed meteorological data which are difficult, if not impossible, to obtain operationally.

The Atmospheric Filter uses sensor registration techniques to leverage existing Link-16 navigation solutions among cooperating platforms to improve range estimates. Refraction model parameters can be estimated as random variables within an Extended Kalman Filter (EKF). These parameters are reasonably constant, or at most slowly varying in the vicinity of the platform on which the algorithm is hosted. Convergence of the filter at a rate faster than these parameters change ensures that accurate real time estimates of key atmospheric parameters are maintained. Our goals in this nine month study are to demonstrate the potential for reducing range estimation errors to the order of one meter at up to 300 km true range, and a few meters for ranges longer than 300 km.

The satisfaction of the second requirement, that of modeling of the signal propagation dynamics, will require consideration of propagation models beyond the "flat earth" model. Analysis of propagation using ray tracing and Snell's Law allow for the development of a "round earth" propagation model. Thus, there are two varieties of Atmospheric Filters built around these two concepts, namely Flat Earth (FE), and Round Earth (RE) designs. As will be described in detail, both of these will be shown to be useful for different flight geometries. The derivation of working equations for these two designs, as well as extensive simulation of expected performance, is presented in later sections of this report.

5.1.1 Sensor Registration Applied to Atmospheric Calibration Processing

The fundamental technology upon which this project is based is that of Sensor Registration, a process developed at BAE SYSTEMS during the past ten years, and which is now being applied to data registration for generating a reliable Single Integrated Air Picture (SIAP) for the US Navy and the Ballistic Missile Defense Organization (BMDO). The basis of sensor registration is the exploitation of Link-16 hybrid navigation within a given community to estimate measurement biases and align sensor axes to common WGS-84 coordinates. The sensors for which this process has been applied include AEGIS SPY-1 Radar, PATRIOT Radars, and similar long range tracking systems.

The basic process of sensor registration is illustrated in Figure 1. In this example, two Link-16 community members are assumed to have performed navigation registration, that is, establishment of correlated position, velocity, and azimuthal alignment to WGS-84 coordinates such that their relative positions and orientations with respect to each other are known to high precision. If both are processing GPS observations, they can achieve 10 meters relative accuracy, or better. By comparing PPLI reported position with sensor measurements, measurement biases and sensor misalignments can be separated and estimated via standard Kalman Filter processing techniques. The key element is that of a measurement standard independent of the sensors themselves.

Sensor Registration Concept
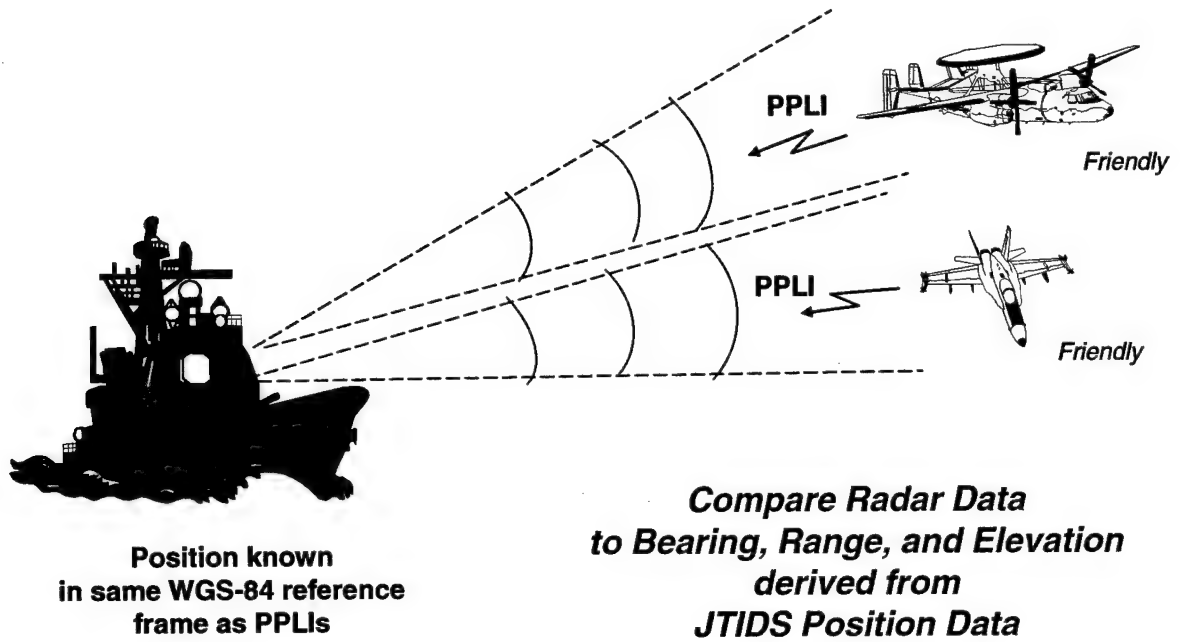
Figure 1 Sensor Registration Concept

AF Operational Description
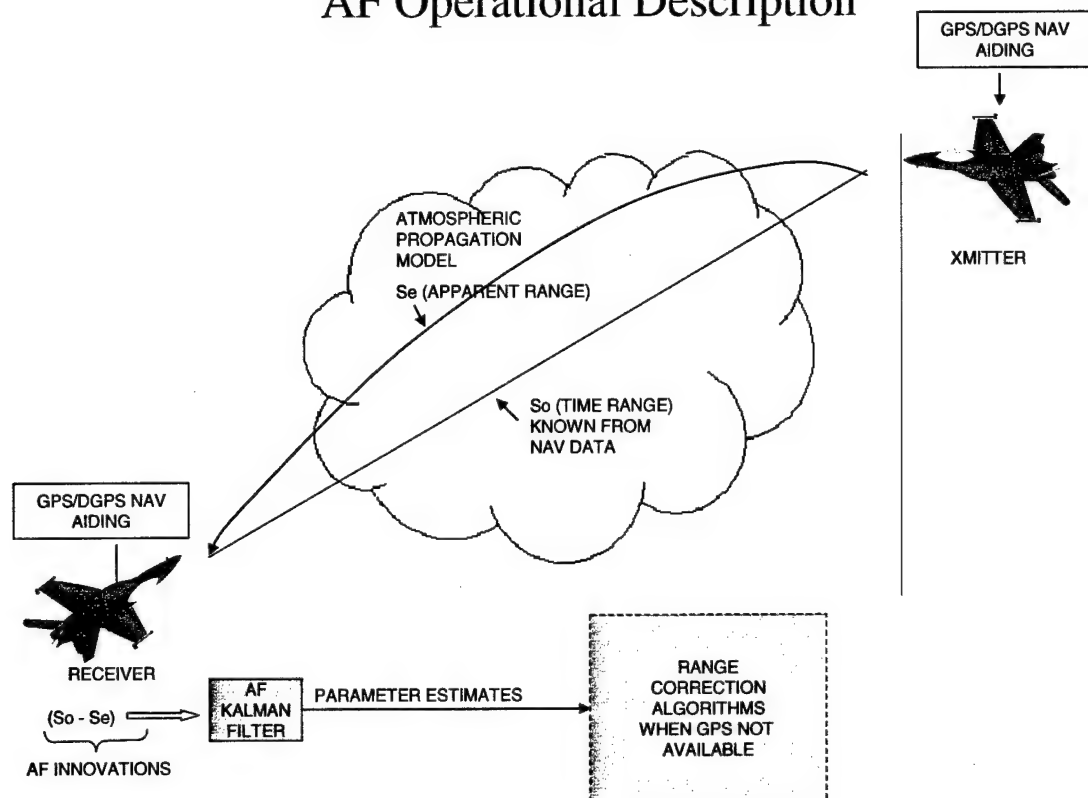
Figure 2. AF Operational Description

The application of this technology has been a resounding success in field trials, calibrating radar biases in azimuth and elevation to submilliradian levels, and range biases to the 20 foot level.

Realizing that a Link-16 terminal measuring range is also a "sensor", the application of this technology to the Atmospheric Filter becomes a straightforward proposition (Figure 2). Assuming that both transmitter and receiver are GPS or DGPS-aided, the measured TOA can then be compared with the predicted TOA, an exact parallel to that described above for radar calibration. The only difference is the replacement of sensor error models with one of two atmospheric models used to predict the key parameters. As will be noted later in this report, the specific atmospheric parameters modeled will be slightly different for Flat Earth (FE) and Round Earth (RE) atmospheric models.

5.1.2 Refraction Modeling

In general, the index of refraction, n, is not known along the propagation path, and thus a model of the atmospheric parameters that determine n is required. Often, an equivalent quantity, termed the refractivity, N, is used, where

$$N = 10^6 * (n-1). \tag{1}$$

This simplifies the discussion in that one deals with numbers like N = 300 instead of n = 1.0003, for example.

There are numerous refractivity models of various complexity ranging from a linear or exponential variation of n with altitude [Bean and Thayer, 1959, Reference 6], to models which give average values of the atmospheric parameters as functions of location (latitude) and time of year along with altitude [Collins and Langley, 1996, Reference 1] . We extract the following Table (Table 1) from Coleman [Coleman, 2000, Reference 4] as an example of typical values of refractivity for different values of pressure, temperature, and relative humidity.

The wide swings in refractivity over the range of values in this table are notable. A conclusion which may be reached from this data is that detailed measurements of pressure, temperature, and humidity would be required to apply the more complex models to predict N. The obvious difficulty of obtaining real time data across wide geographic areas leads us to seek out simpler models for which we can apply the principles of sensor registration. It will be demonstrated that such models exist, and, using sensor registration to estimate key parameters, can predict range errors with equivalent accuracy to the more complex models.

P (mbar)	T (K)	RH (%)	N_{hyd}	N_{wet}	N	Comments
1013.25 (29.92 in)	288.15 (59 °F)	69	272.9	52.5	325.4	Global Average Surface Values
1012.9 (29.91 in)	280.4 (45 °F)	92	280.3	44.3	324.7	NYC, 1/10/00, 10 AM
1023.4 (30.22 in)	258.2 (5 °F)	60	307.6	6.4	314.4	NYC, 1/18/00, 8 AM
1006.1 (29.71 in)	266.0 (19 °F)	43	293.6	8.1	301.7	NYC, 1/21/00, 4 PM
1020.3 (30.13 in)	292.6 (67 °F)	23	270.6	22.6	293.2	NYC, 5/16/00, 4 PM
1013.25 (29.92 in)	303.15 (86 °F)	15	259.4	25.8	285.2	Hot and Dry
1013.25 (29.92 in)	303.15 (86 °F)	90	259.4	155.0	414.4	Hot and Humid

Table 1 Refractivity, N, for some different atmospheric conditions.

Shown are some selected values of pressure (P), temperature (T) and relative humidity (RH). Also, N_{hyd} is the refractivity due to the natural gases of the atmosphere and N_{wet} is that due to water vapor, $N = N_{hyd} + N_{wet}$.

5.1.3 Linear n and Exponential Models

The following text is reproduced in large part from Coleman (Reference 4).

5.1.3.1 Linear Index of Refraction Model

Probably the most common index of refraction model, used for low precision line of sight propagation calculations, is the "effective earth radius" model, where the actual earth radius is replaced by an "effective" earth radius

$$R_e' = kR_e, \qquad (2)$$

where k is the effective earth's radius factor, typically about 4/3 (but can vary from 0.5 to 4.0) and Re is the true radius of the earth (equaling about 6378 km). The radio rays are then assumed to propagate as straight lines on this enlarged earth. The effective earth radius factor is given by

$$k = \frac{1}{1 + \dfrac{R_e + h}{n} \dfrac{dn}{dh}}, \qquad (3)$$

where h = the height in meters above mean sea level, for rays leaving the antenna at low elevation angles. Since k is assumed constant and n is nearly equal to one, dn/dh is effectively being modeled as a constant. Even though this is untrue, the model works well.[4] Solving for dn/dh with k = 4/3 yields $dn/dh = -39x10^{-9} / m$, which is approximately the average change in the refraction value in the first kilometer of the atmosphere. Although this model is too crude for use by Link-16, it will prove useful in improving the current Link-16 algorithm, as will be discussed subsequently.

5.1.3.2 Exponential Refractivity Model

The model underlying the current Link-16 compensation algorithm is an exponential variation with altitude[5]

$$n = 1 + a\,e^{-bh} \quad (\text{or } N = N_0\,e^{-bh} \,, \; N_0 = a \times 10^6) \qquad (4)$$

---

[4] Bean, B.R. and Thayer, G.D., "Models of Atmospheric Radio Refractive Index," *Proc. Of the IRE*, May 1959, pp. 740-755. (Reference 6).
[5] Singer Kearfott, Internal Document Y258A002 (Reference 7), and Bean, B.R. and Thayer, G.D., op cit.

which are reasonably close to the values in (5). Based on a least squares fit to refraction values over 0 to 15 km, calculated from global average values of atmospheric parameters, as discussed below, the nominal values of a and b used here will be

$$a = 326.6 \times 10^{-6}, b = 0.130061/\text{km and} \qquad (8).$$

Alternately, the same procedure that was used to derive the values in (7) could be followed with, say, $N_S = 325$. Then $a = 321.6 \times 10^{-6}$, $b = 0.1244/\text{km}$, which are reasonably consistent with the values chosen in (8).

Given a and b, the effective earth radius factor k can be calculated as $k = 1/(1 - ab\,R_e)$. Doing this for the two sets of a, b parameters in (5) gives k = 1.184 and 1.562, respectively, while the same calculation for the value in (8) yields k = 1.374.

5.1.4 Comparison of Atmospheric Refractivity Models

In searching for the refractive model most suitable for adaptation to the Atmospheric Filter, we considered, in addition to the exponential model, two very widely accepted refractivity models, the Smith-Weintraub and Thayer models, described below.

The exponential model is a two-parameter model which depends upon altitude and the other two (discounting known constants) are three parameter models which depend upon meteorological data. The question we wish to answer is if, within the ranges of interest, the simpler exponential model can provide a good fit to the more complex models. The result, shown in Figures 3 – 9, below, is that the exponential model does provide a good fit to these, with the proper estimation of its two parameters. *What this means is that a decaying exponential is generally a good fit to the behavior of the atmospheric refractivity, as a function of altitude.* A Kalman Filter is therefore a very good choice for estimating its parameters, as they are random constants.

The atmospheric refractivity models examined are:

1) The Exponential Model[6], refractivity $N = a\,e^{-bh}$ (equivalently, $n = 1 + 10^{-6} a\,e^{-bh}$), where a and b are constants for a given set of atmospheric conditions and h = altitude. The parameter a is the refractivity at h = 0 and b is the decay constant.

2) The Smith-Weintraub Model[7], $N = 77.6\,(P/T) + 3.73 \times 10^{5}\,(e/T^{2}) = N_{hydrostatic} + N_{wet}$, where P is total atmospheric pressure (mbar), T is the temperature in degrees Kelvin, and e is the partial pressure of water vapor (mbar). $N_{hydrostatic}$ is the refractivity due to the natural gases of the atmosphere and N_{wet} is that part due to the dipolar part of water vapor[8]. The Smith-Weintraub Model is sometimes written in the equivalent form $N = 77.6\,/T\,(P + 4810.0\,e/T)$. A similar refractivity

[6] Singer Kearfott, Internal Document (Reference 7) and Coleman (Reference 4).
[7] Flores, A., Ruffini, G. and Rius, A., op. cit., pp. 223-224.
[8] Flores, A., Rius, A., et al. "Sensing Atmospheric Structure: Tropospheric tomographic results of the small-scale GPS campaign at the Onsala Space Observatory," *Earth, Planets, Space*, **52**, 941-945, 2000, p. 941.

model, the Saastamoinen Model, was also examined[9]. It was found to give results that were very similar to those of the Smith-Weintraub model and so is not discussed here. This is mentioned for completeness.

3) The Thayer Model[10], $N = K_1 (P_d / T) Z_d^{-1} + [(K_2 (e/T) + K_3 (e/T^2)] Z_W^{-1}$. Where P_d is the partial pressure of the dry gases in the atmosphere (mbar), e is the partial pressure of the water vapor (mbar), T is the absolute temperature (degrees Kelvin), Z_d is the compressibility factor for dry air, approximately 1, Z_w^{-1} is the inverse compressibility factor for moist air[11], and the K_i are empirically determined and well known constants. For reference,
$$Z_w^{-1} = 1 + 1650\left(e/T^3\right)\left(1 - 0.01317t + 1.75x10^{-4}t^2 + 1.44x10^{-6}t^3\right),$$ where t equals the temperature in degrees Celsius. A variant of the Thayer refractivity model, the so-called Davis-Thayer Model, as modified by Davis, et al. in 1985[12], was also examined. It was found to give results that were almost identical to those of the Thayer model and so is not discussed here. This is mentioned for completeness.

In the last two models, the pressures and temperatures are needed as a function of altitude for a given time of day, region of earth and season. In these, the parameter e, the partial water vapor pressure, is obtained[13] from the available relative humidity data as follows. To calculate vapor pressure (e) in mbar (millibars) from the temperature in degrees Kelvin and relative humidity (RH), divide RH by 100 to convert from percent to a fraction and multiply by $e_s(T)$. Thus,

$$e = e_s(T)\left[\frac{RH}{100}\right],$$

where

$$e_s = 0.61121 \exp\left(\frac{17.502(T - 273.15)}{T - 32.18}\right).$$

The estimates of N from these three models were compared to each other under a wide range of environmental factors, these results are discussed below.

[9] Saastamoinen, J., "Contributions to the Theory of Atmospheric Refraction," *Bulletin Geodesique*, Nos. 105, 106, and 107, pp. 279-298 and pp. 13-34.
[10] Thayer, D., op. cit., pp. 803-807; and Collins, J.P. and Langley, R. B., Reference 1, Equation (2).
[11] Owens, J.C., "Optical Refractive Index of Air: dependence on pressure, temperature and composition," *Appl. Opt., 6*, 51-58, 1967.
[12] Davis, J.L., T.A. Herring, I.I. Shapiro, A.E.E. Rogers, and G. Elgered, "Geodesy by radio interferometry: Effects of atmospheric modeling errors on estimates of baseline length," *Radio Sci., 20*,1593-1607, 1985.
[13] Federal Meteorological Handbook No. 3 – Rawindsonde and Pibal Observations. FCM-H3-1997, May, 1997.

The meteorological data came from the University of Wyoming's Department of Atmospheric Science on-line database[14]. This web site has an extensive database of actual atmospheric meteorological soundings from 1989 to the present for sites all over the world, for all months of the year and various times of day, for altitudes from sea level up to about 30 km. The text of the data used to generate part of Figure 3 for December 2002, for example, is shown in Appendix A.

From this database, air pressure, altitude, temperature and relative humidity data from a selection of sites from over the Polar Regions, the Tropics, deserts, temperate land masses and oceans was used to calculate the Smith-Weintraub and Thayer refractivities. For these locations, atmospheric sounding data from different seasons of the year and different times of day was selected to cover extremes of climatic conditions and daily variations.

These results confirm the very similar behavior of the Smith-Weintraub and Thayer models and that the refractivity given by the Exponential model agrees well with them.

Figure 3 shows a comparison of the three atmospheric refractivity models' results for actual University of Wyoming meteorological data for Goteborg, Sweden from two times, twelve hours apart, on 4 December, 2002. Results from July of that year are also included for contrasting data from the summer. The Smith-Weintraub and Thayer models give nearly identical results (their curves match so well that their difference almost cannot be seen in the figure) and the Exponential model (solid curve) agrees well with them.

Figures 4 through 10 display results for the three models from parts of the earth chosen for diverse climates and from different times of the year 2002 (Hyderabad, India; Beijing, China; Midland, Texas; and Davis, Antarctica). Again, as in the Goteborg results, the exponential model gives a good fit to the Smith-Weintraub and Thayer models of refractivity at altitudes of operational interest (5 km and higher) and the results of the latter two models match closely. In Figures 7 and 8, one can note a tiny difference in the results of the Smith-Weintraub and Thayer models at low altitudes.

Figure 10 shows results for Brisbane, Australia on 1200 GMT August 4, 1997. These results are included as there are measurements of the refractivity available[15] for Brisbane at that time at sea level, the average for that day is N = about 338. Our results are in agreement with that.

The parameters of the Exponential model used to obtain the curves in Figures 3 through 10 are given in Table 2. They were extracted from a least squares two parameter exponential curve fit to the data using MATLAB, a high-performance environment for technical computing. As a Kalman Filter gives a very robust least squares type fit, this is a good indicator of how well the AF would model refractivity. The refractivity coefficient of height, b, is exactly as given by the program. The zero altitude parameter a was adjusted

---

[14] University of Wyoming, College of Engineering, Department of Atmospheric Sciences, *Upper Air Meteorological Soundings Website*. URL: http://weather.uwyo.edu/upperair/sounding.html
[15] "Australian Amateur Radio Microwave Expedition."
http://members.ozemail.com.au/~tecknolt/ausmicro/dxpedit.htm , Figure 4.

slightly, from that given by the program, translating the entire curve each time, in order to give a better fit at altitudes of over 5 km. Thus, Table 2 shows the parameters for the curve fit that would be obtained using data from 5 km and higher.

Locality	Date (GMT)	a (= N at sea level) (dimensionless) (approximate value)	b (per m) (x 10^{-4})
Goteborg, Sweden	12Z 04 Dec,2002	340	1.4090
Goteborg, Sweden	00Z 04 Dec,2002	340	1.3642
Goteborg, Sweden	12Z 04 July,2002	340	1.3946
Hyderabad, India	12Z 04 July,2002	350	1.3293
Kwajalein Island, Pacific Ocean	12Z 04 Dec,2002	380	1.4150
Beijing, China	12Z 04 Dec,2002	350	1.3454
Bermuda	00Z 04 Dec,2002	330	1.2806
Midland, Texas	12Z 04 Dec,2002	360	1.4660
Brisbane, Australia	12Z 04 Aug,1997	330	1.3021
Davis, Antarctica	12Z 04 Dec,2002	320	1.4508

Table 2 Exponential Model Parameters Used to Fit Data in Figures 4 to 9 (Z = Universal Time, GMT)

Refractivity at 00Z04Dec2002
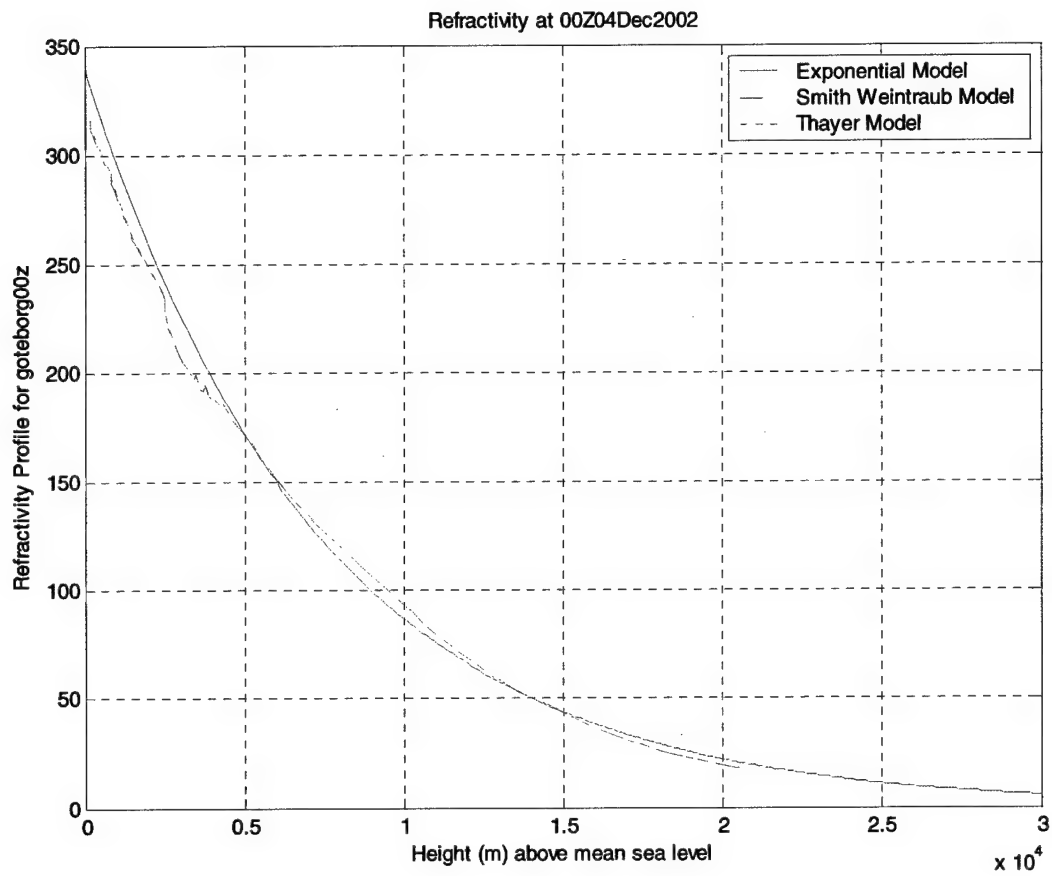
Modeled Refractivities From Actual Meteorological Data (from Goteborg, Sweden) at 0000 Hours GMT on 4 December, 2002.
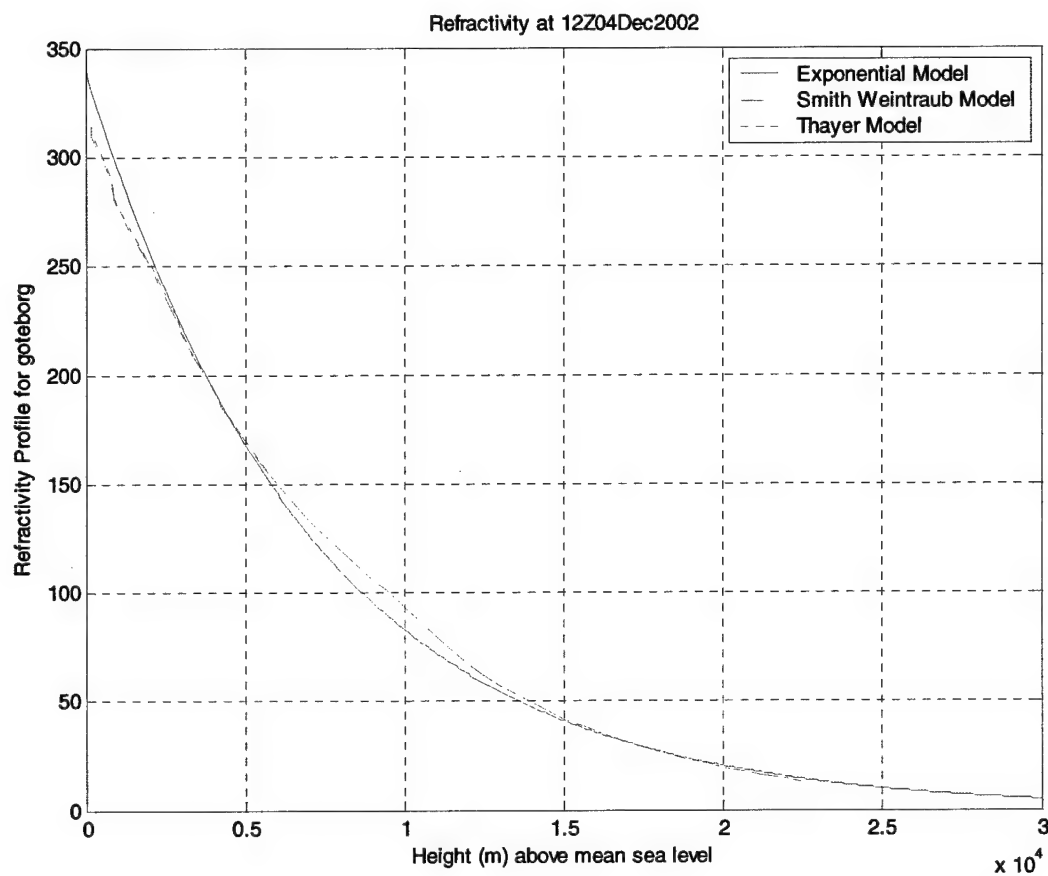
Modeled Refractivities From Actual Meteorological Data (from Goteborg, Sweden) at 1200 Hours GMT on 4 December, 2002.
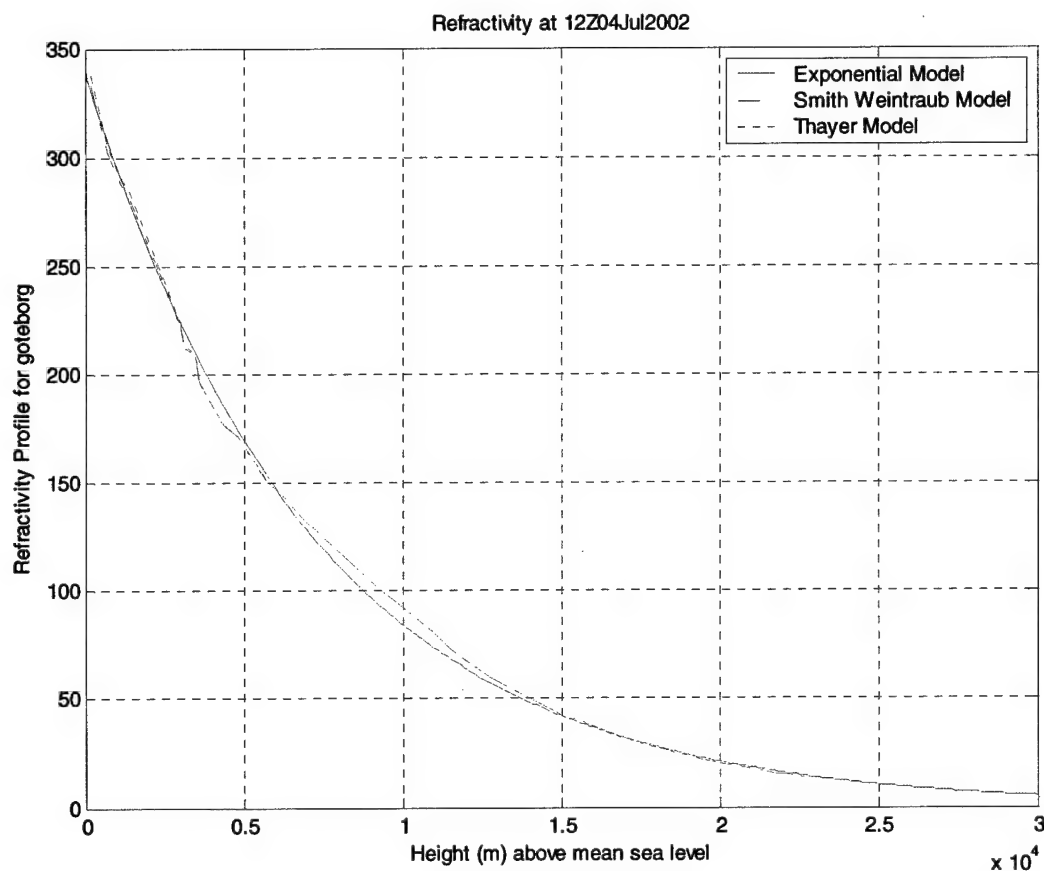
Modeled Refractivities From Actual Meteorological Data (from Goteborg, Sweden) at 1200 Hours GMT on 4 July, 2002.

Figure 3 Modeled Refractivities From Actual Meteorological Data (from Goteborg, Sweden) from July and December, 2002.
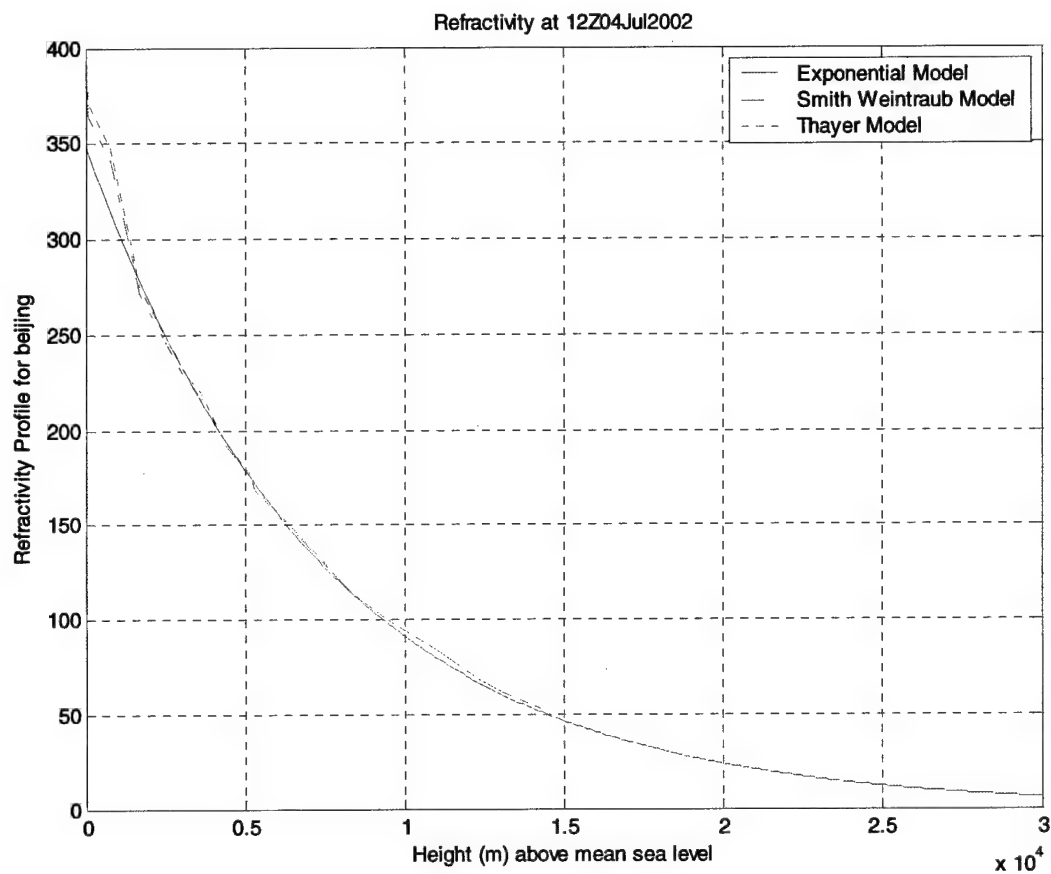
Figure 4 Modeled Refractivities From Actual Meteorological Data (Beijing, China).
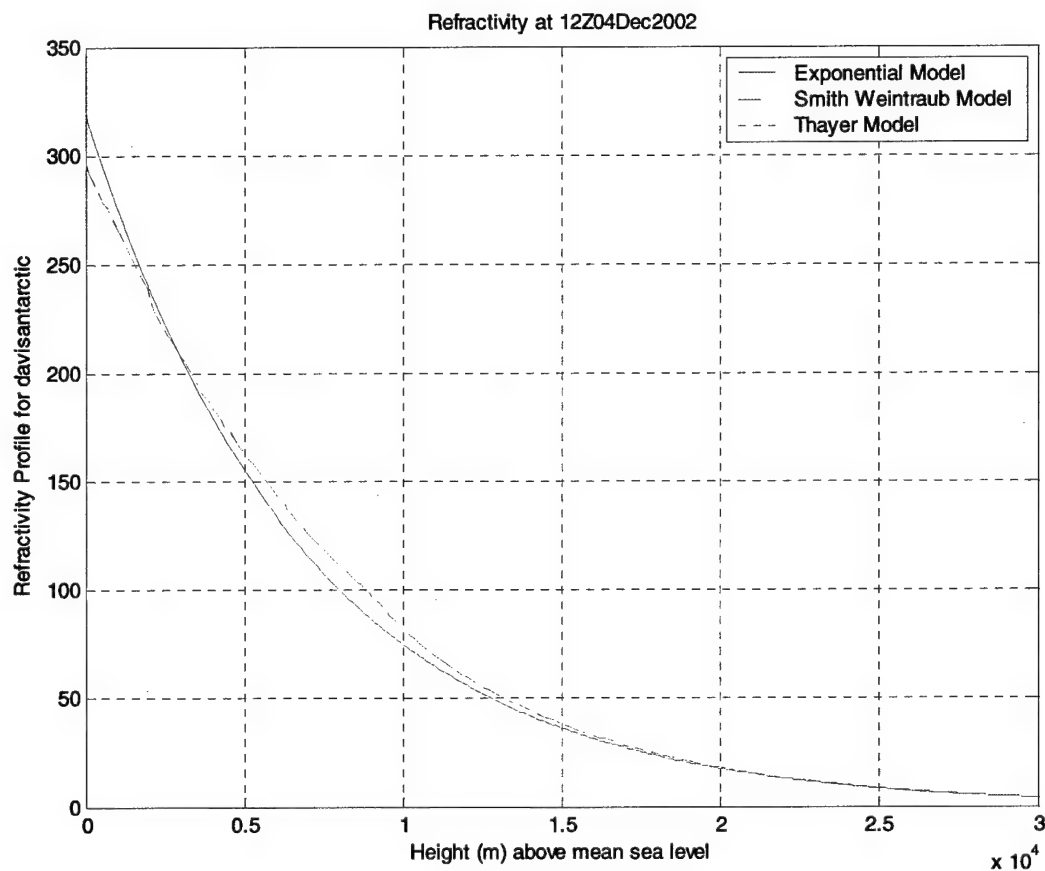
Figure 5 Modeled Refractivities From Actual Meteorological Data (Davis Station, Antarctica).
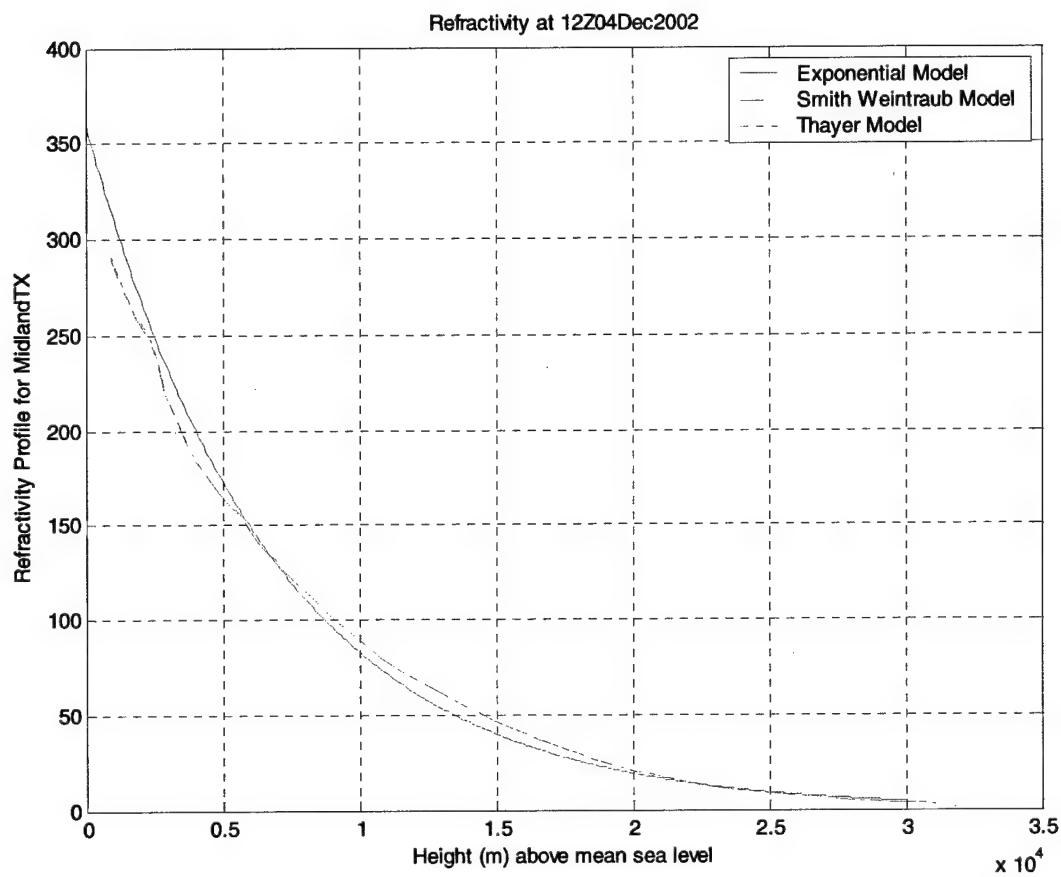
Figure 6 Modeled Refractivities From Actual Meteorological Data (Midland, Texas).
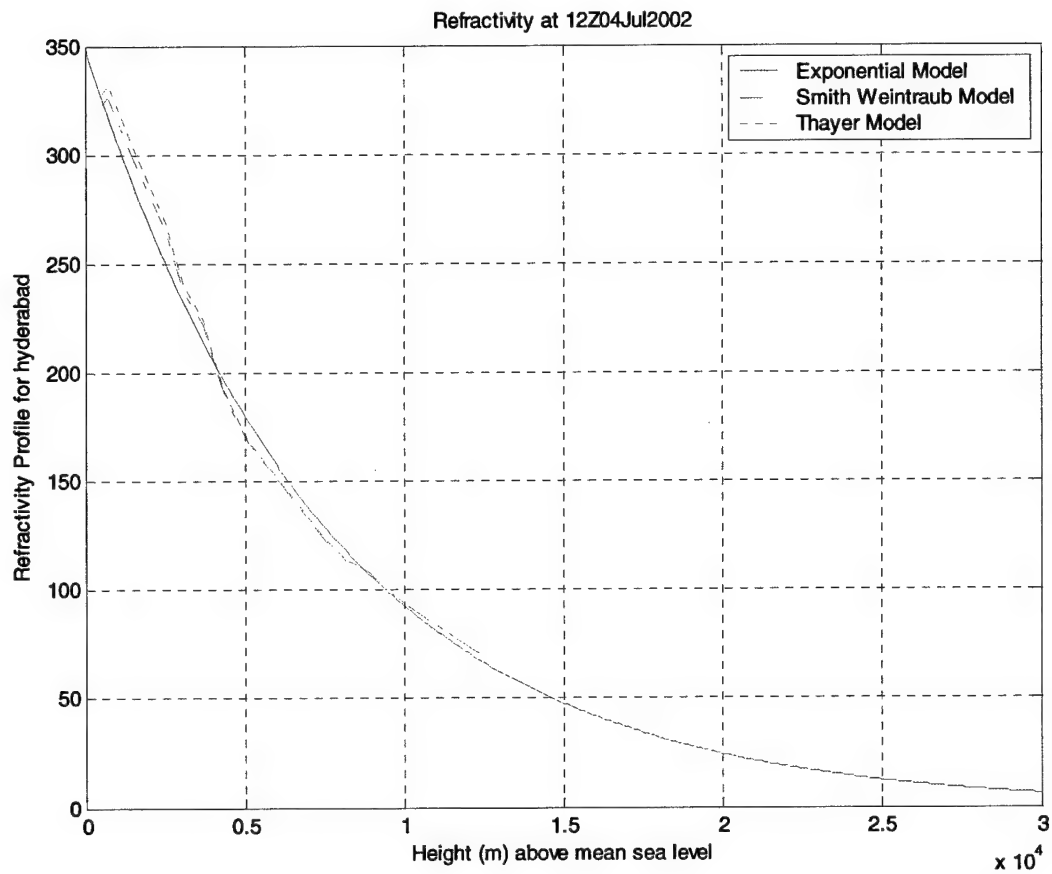
Figure 7 Modeled Refractivities From Actual Meteorological Data (Hyderabad, India).
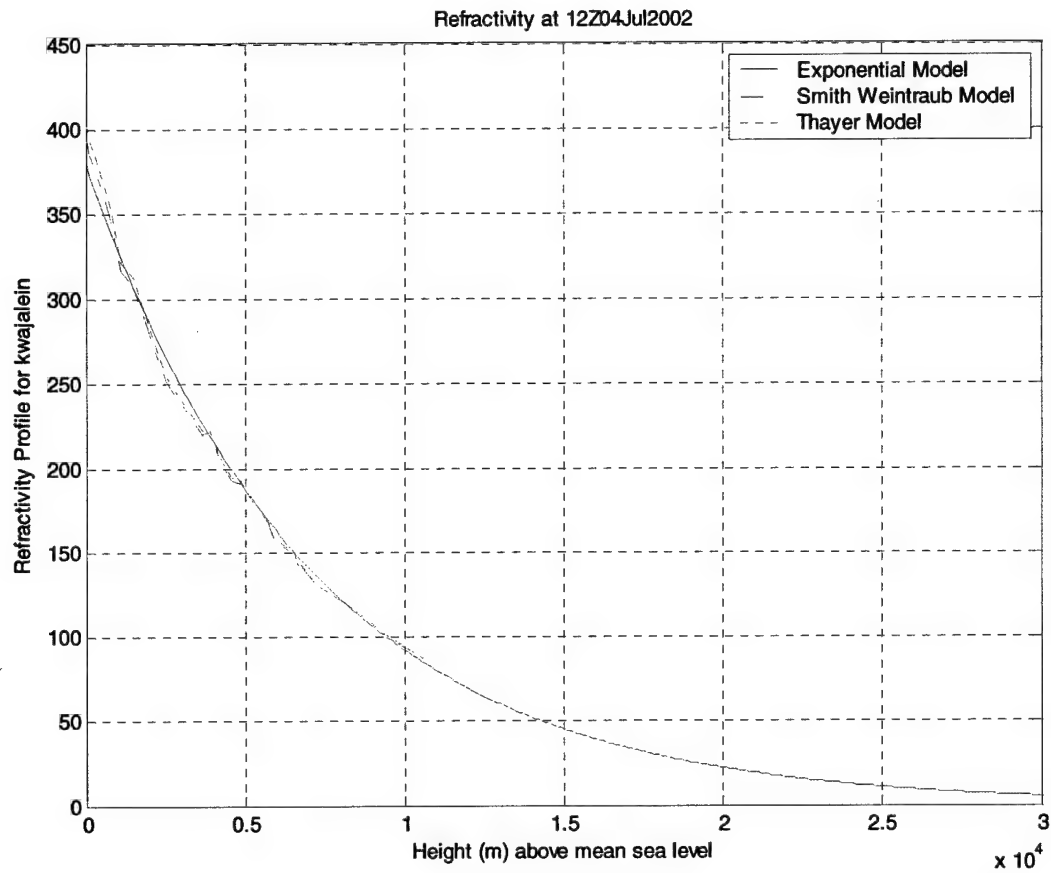
Figure 8 Modeled Refractivities From Actual Meteorological Data (Kwajalein Island, Pacific Ocean).
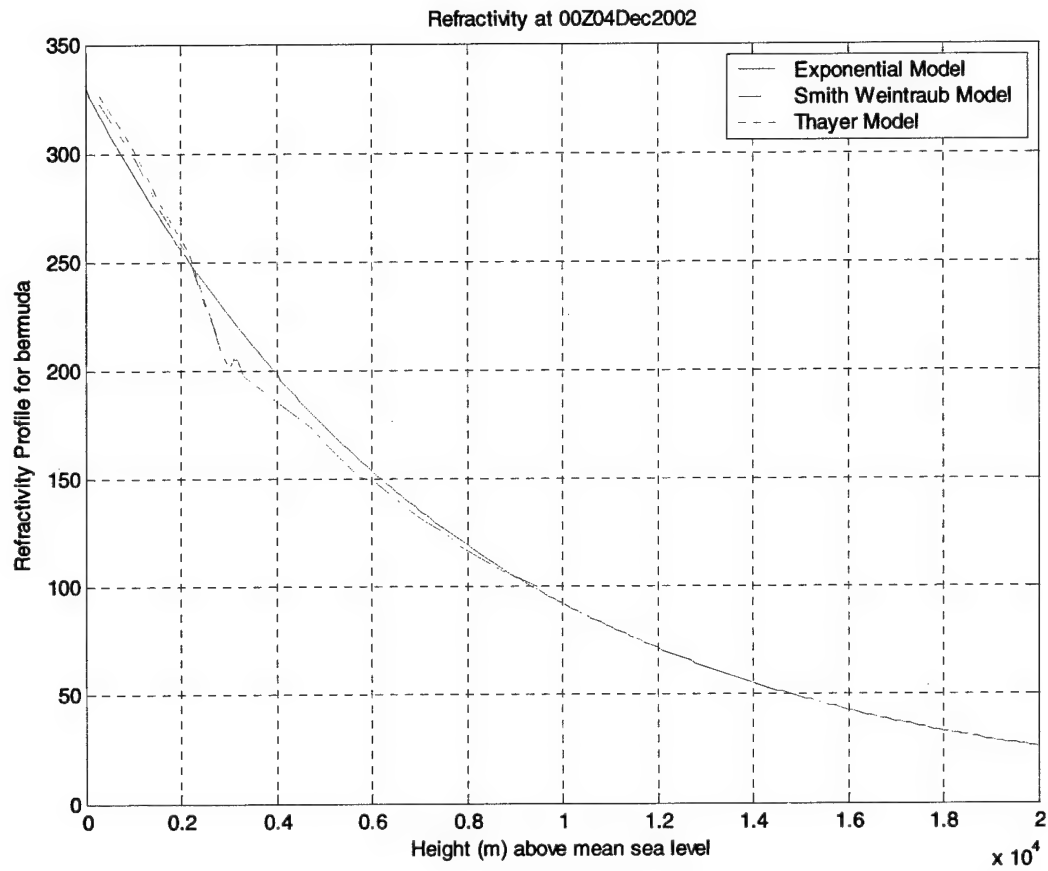
Figure 9 Modeled Refractivities From Actual Meteorological Data (Bermuda, Atlantic Ocean).
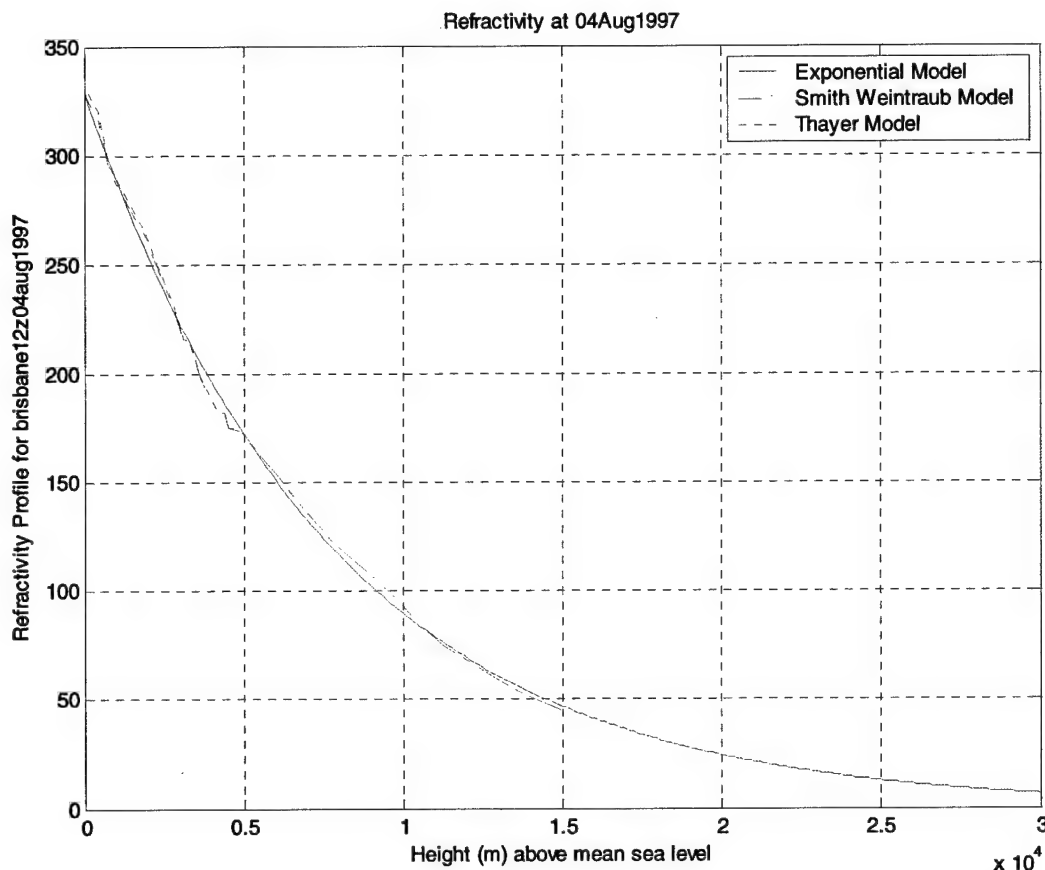
Figure 10 Modeled Refractivities From Actual Meteorological Data (Brisbane, Australia).

Figures 4 through 10 show comparisons of the three refractivity models for July and December 2002. Figure 10 also shows results for Brisbane, Australia for August 4, 1997 for which there are published measured refractivity results and which show good agreement, see below.

As can be seen, the results are quite similar for all the refractivity models under all the environmental extremes examined herein. Thus, *there is no reason to use either the Smith-Weintraub or Thayer model* since the Exponential model is not only simpler but also performs well at altitudes of operational interest as compared to the more complicated models requiring meteorological data which would be difficult to obtain operationally and which also would require additional processing and database management.

In the test bed simulations run to exercise the AF, the same exponential model for the Kalman Filter state vector is used to estimate parameters a and b and using refractivities from both the Smith-Weintraub and Exponential models to generate the range results. The Thayer model was not used as Figures 3 through 10 show that its results are extremely similar to those of the Smith-Weintraub model. A signal from an emitter at a known range to the receiver (truth range) was simulated. The Time of Arrival (TOA) of the signal was

calculated as if the signal were refracted over the truth range per the Smith-Weintraub or Exponential models. The estimated refractivity parameters were then used to calculate the range estimate as will be explained in a later section of this report. The truth range was subtracted from this to generate the range errors. The results for the two different AF models of the structure of the atmosphere, the FE or Flat Earth model and the RE or Round Earth model, also to be described below. As one would expect from Figures 3 through 10, the results of the AF using either the Exponential Model or the Smith-Weintraub Model turn out to be quite similar, as will be shown later in this report.

As the Exponential Model of atmospheric refractivity works well, is simpler and is only a function of altitude, it should be used operationally in the AF. It wins the competition, hands down.

5.2 Atmospheric Refraction: Snell's Law and Ray Tracing Dynamic Equations

Coleman (Reference 4) presents an illuminating analysis of the application of Snell's Law to both flat earth and round earth models. This analysis provides rigorous definitions and derivations for the critical error components which will be denoted by the terms Excess Path Length and Curvature Correction in both domains. Furthermore, Coleman describes an ingenious numerical procedure for solving ray path trajectory elements and thus provides a "truth model" against which our compensation algorithms, yet to be derived, can be compared.

Because of the importance of these equations to the derivation of the ultimate compensation algorithms, this section from his report is hereby reproduced in its entirety.

5.2.1 Atmospheric Refraction Effects with Application to Link-16 TOA Compensation

It is assumed that ray theory is adequate to describe the message propagation and that the propagation occurs in a plane. Figure 8 illustrates the situation and defines the pertinent quantities of interest. The transmitting terminal is located at P_0 at an altitude h_0; the receiving terminal is located at P_1 at an altitude of h_1. The straight-line path between P_0 and P_1 is denoted by C_0 and has length S_0 and elevation angle E with respect to the local horizontal at P_0. The actual propagation path of the ray is curved and is denoted by C and has a length S.

The tangent to the ray path at P_0 makes an angle θ_0, the ray's local elevation angle at P_0, with the local horizontal at P_0. This is the angle an observer at P_0 would measure as the elevation angle of P_1 and is called the apparent elevation angle. In the absence of ray curvature, the ray's local elevation angle at P_1, θ_1, would simply be $\theta_0 + \phi$ due simply to earth curvature. However, due to ray curvature, $\theta_1 = \theta_0 + \phi - \beta$, where β, the ray's curvature or "bending" parameter, is the angle between the tangents to the ray path at P_0 and P_1.
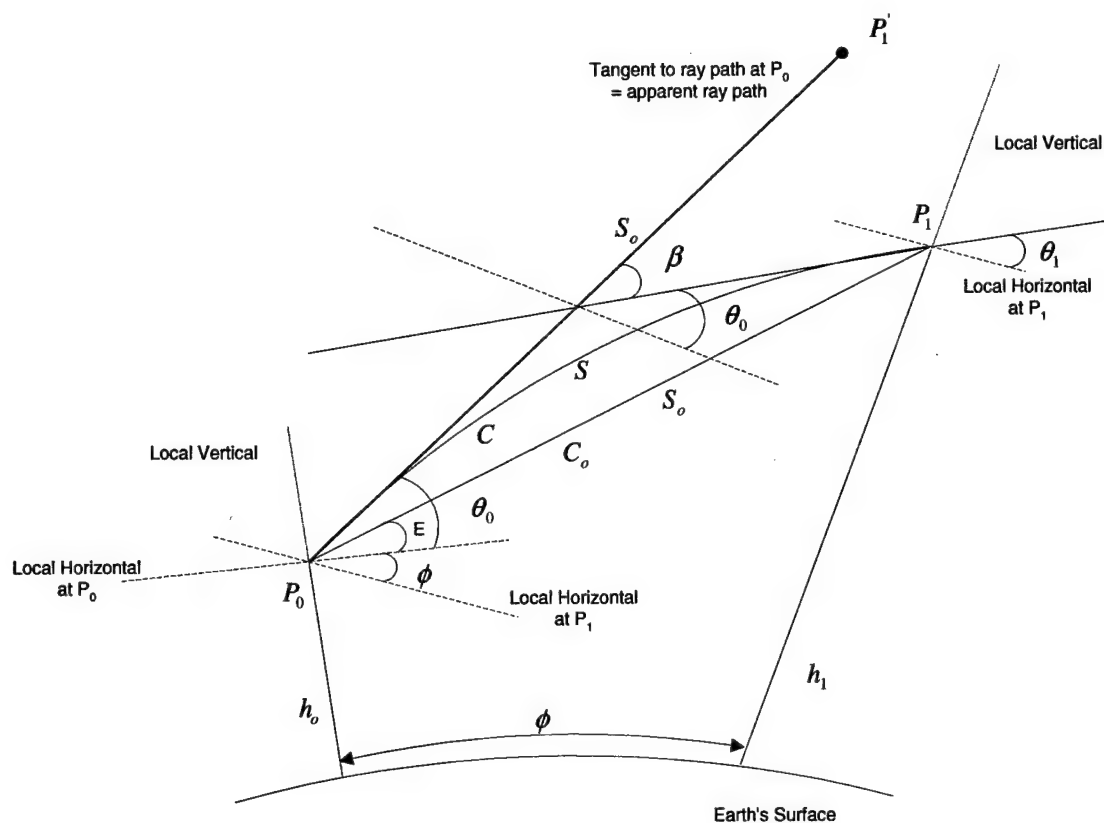
Figure 11 General Effects of Ray Propagation

The propagation time of the signal, the TOA, will be denoted by τ. In a vacuum or a medium of constant index of refraction, the ray would travel along C_0 and $\tau = S_0/(c/n))$. In the atmosphere, the ray propagates along the curved path shown since this path yields the smallest propagation time of all paths between P_0 and P_1 (Fermat's principle, of which Snell's Law is an expression). The explanation is that the curved path allows the ray to reach the region of smaller index of refraction sooner than the straight line path does, and the gain in propagation speed more than compensates for the longer curved propagation path.

To travel the small distance ds requires an interval of time $dt = ds/ c_{air} = (n/c)$ ds, thus the signal's propagation time is given by

$$\tau = \int_c dt = 1/c \int_c n \, ds \qquad (9)$$

The effective (or electrical or "optical") path length or apparent range is defined as

$$S_e = \int_c n\, ds = c\,\tau \tag{10}$$

Thus Se and τ are equivalent parameters differing only by a known scale factor. With the geometric length of the ray path given by

$$S_0 = \int_c ds \tag{11}$$

it proves convenient to define the quantity

$$L_e = S_e - S = \int_c (n-1)\, ds \tag{12}$$

which is called the "excess path length." The length of S_0 is given by

$$S_0 = \int_{c_0} ds \tag{13}.$$

Notice the difference between (11) and (13). The curvature correction L_c is defined as

$$L_c = S - S_0 = \int_c ds - \int_{c_0} ds \;. \tag{14}$$

Then the total correction L, the correction that allows S0 to be calculated from the measured Se is

$$L = L_e + L_c = S_e - S_0 = \int_c n\, ds - \int_{c_0} ds \tag{15}$$

The curvature correction L_c is much smaller than the excess path length Le and often can be neglected.

5.2.1.1 Ray Propagation – Flat Earth

Figure 4 illustrates ray propagation for a "flat Earth" model. The atmosphere is modeled
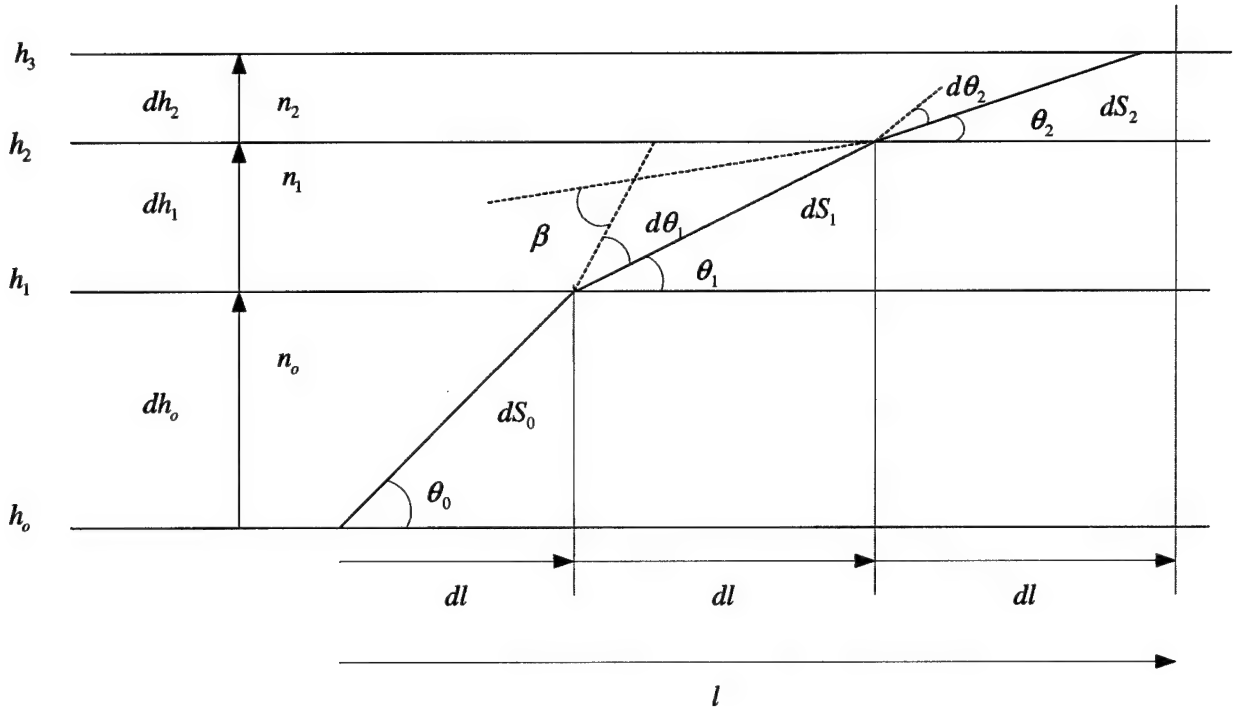
Figure 12 Flat Earth Refraction Model

as being stratified into discrete flat layers. The actual ray path is modeled as a series of chords. Snell's law written in terms of the elevation angle θ yields

$$n_{i+1} \cos\theta_{i+1} = n_i \cos\theta_i \tag{16}$$

The horizontal increment dl is assumed given and small enough such that the change in θ and n is small and thus (16) can be approximated as

$$d\theta_i = \theta_{i+1} - \theta_i \approx \cot\theta_i (n_{i+1} - n_i)/n_i = \cot\theta_i \, dn_i/n_i \tag{17}$$

where $dn_i = n_{i+1} - n_i$. Since

$$dh_i = dl \tan\theta_i \tag{18}$$

(17) can be expressed as

$$d\theta_i = \cot\theta \, dn_i/dh_i \, dh_i \cot\theta_i = (dn_i/dh_i) \, dl/n_i \tag{19}$$

which eliminates the singularity when $\theta_i = 0$. The geometric and effective lengths of the ray path are calculated via

$$dS_i = dl/\cos\theta_i ,$$
(20)

$$dSe_i = n_i \, dl/\cos\theta_i ,$$
(21)

In the limit, as dl approaches zero,

$$dl/d\theta = \frac{dn/d\theta}{n},$$
(22)

$$dh/dl = \tan\theta,$$
(23)

$$dS/dl = 1/\cos\theta,$$
(24)

$$dS_e/dl = n/\cos\theta,$$
(25)

$$\beta = \theta_0 - \theta \quad (\text{or } d\beta = -d\theta)$$
(26)

These equations are solved (approximately) via the discrete approximations in (18) – (21).

Specifically, starting with a given h_0 and θ_0, dh_0 is calculated via (18), dn/dh is evaluated at h_0 and then θ_0 is calculated via (19), etc. Thus

$$l = dl + dl + dl + ...$$

$$\theta = \theta_0 + d\theta_0 + d\theta_1 + d\theta_2 + ...$$

$$h = h_0 + dh_0 + dh_1 + dh_2 + ...$$

$$S = S_0 + dS_0 + dS_1 + dS_2 + ...$$

$$S_e = n_0 \, dS_0 + n_1 \, dS_1 + n_2 \, dS_2 + ...$$
(27)

It was found that $dl = 10$ m worked well in (18) – (21). Solving these equations given h_0, h_1, and S is much more difficult, it being necessary to solve a two point boundary value problem. In this case, the equations were solved by trial and error, i.e., choose a θ_0 and solve as above, repeat until a solution is found that solves the boundary conditions or not.

Although this model illustrates the effects of refraction nicely, it is insufficient for Link-16 needs. This is due to neglecting the effect the earth's curvature has on the altitude of the ray, and hence on the value of the refraction modeled. This will be discussed after the "round earth" model is developed in the next section. Table 3 tabulates some results from the flat earth model.

l (km)	h (m)	θ (deg)	S (km)	$S_e - S_0$ (m)	$S - S_0$ (m)	β (deg)	E (deg)	N	dN/dh (1/km)	R_r/R_e
0.000	1000.0	0.000	0.000	0.0	0.0	0.000	0.000	286.6	-37.4	4.189
50.00	953.2	-0.107	50.000	14.4	0.0	0.107	-0.054	288.4	-37.4	4.164
100.00	812.1	-0.216	100.002	29.0	0.1	0.216	-0.108	293.8	-38.4	4.088
150.00	575.0	-0.328	150.008	44.0	0.2	0.328	-0.162	303.0	-39.6	3.963
200.00	238.9	-0.443	200.009	59.8	0.5	0.443	-0.218	316.6	-41.4	3.793
0.00	5000.0	0.301	0.000	0.0	0.0	0.000	0.301	170.0	-22.2	7.063
100.00	5415.9	0.177	100.009	16.5	0.0	0.124	0.238	161.0	-21.0	7.457
200.00	5621.1	0.059	200.011	32.5	0.1	0.242	0.178	156.8	-20.5	7.659
300.00	5621.1	-0.059	300.011	48.5	0.5	0.359	0.119	156.8	-20.5	7.659
400.00	5415.9	-0.177	400.014	65.0	1.1	0.478	0.060	161.0	-21.0	7.457
500.00	5000.0	-0.301	500.022	82.6	2.2	0.601	0.000	170.0	-22.2	7.063
0.00	10000.0	0.161	0.0	0.0	0.0	0.000	0.161	88.5	-11.6	13.57
100.00	10223.1	0.095	100.003	8.7	0.0	0.065	0.128	85.9	-11.2	13.97
200.00	10333.9	0.032	200.003	17.3	0.0	0.129	0.096	84.7	-11.1	14.17
300.00	10333.9	-0.032	300.003	25.8	0.1	0.192	0.064	84.7	-11.1	14.17
400.00	10223.1	-0.095	400.004	34.5	0.3	0.256	0.032	85.9	-11.2	13.97
500.00	10000.0	-0.161	500.006	43.6	0.6	0.321	0.000	88.5	-11.6	13.57

Table 3 Flat Earth Model Examples

(Note: refraction estimates in this table are calculated using the exponential model with a = 326.6x10-6 and b = 0.13061 /km and R_r = dS/dθ is the instantaneous curvature of the ray path).

5.2.1.2 Ray propagation – Round Earth

Figures 8 and 10 illustrate the situation for a round earth model. Deriving the modeling equations is very similar to the flat earth case with a few generalizations. The analogue of the horizontal increment dl in the flat earth case is $(R_e + h)\, d\phi$ here. Then (18) becomes

$$dh_i = (R_e + h_i)\, \tan\theta_i\, d\phi \tag{28}$$

Since the elevation angle θ in Figure 3 is defined with respect to the local horizontal, which is continuously rotating with φ, its value represents both ray bending and angular displacement along the surface of the earth. Thus, the introduction of the ray bending parameter or angle β. Since β is defined only with respect to the incident and refracted rays, it satisfies the same equation, within a sign, as θ did in the flat earth case,

$$d\beta_i \approx -\frac{dn_i}{n_i} \cot\theta_i. \tag{29}$$

(Another way to derive (29) is directly from Snell's law for spherical boundary surfaces $n_i (R_e + h_i) \cos\theta_i = n_{i+1}(R_e + h_{i+1}) \cos\theta_{i+1}$ (See references 5 and 6)).
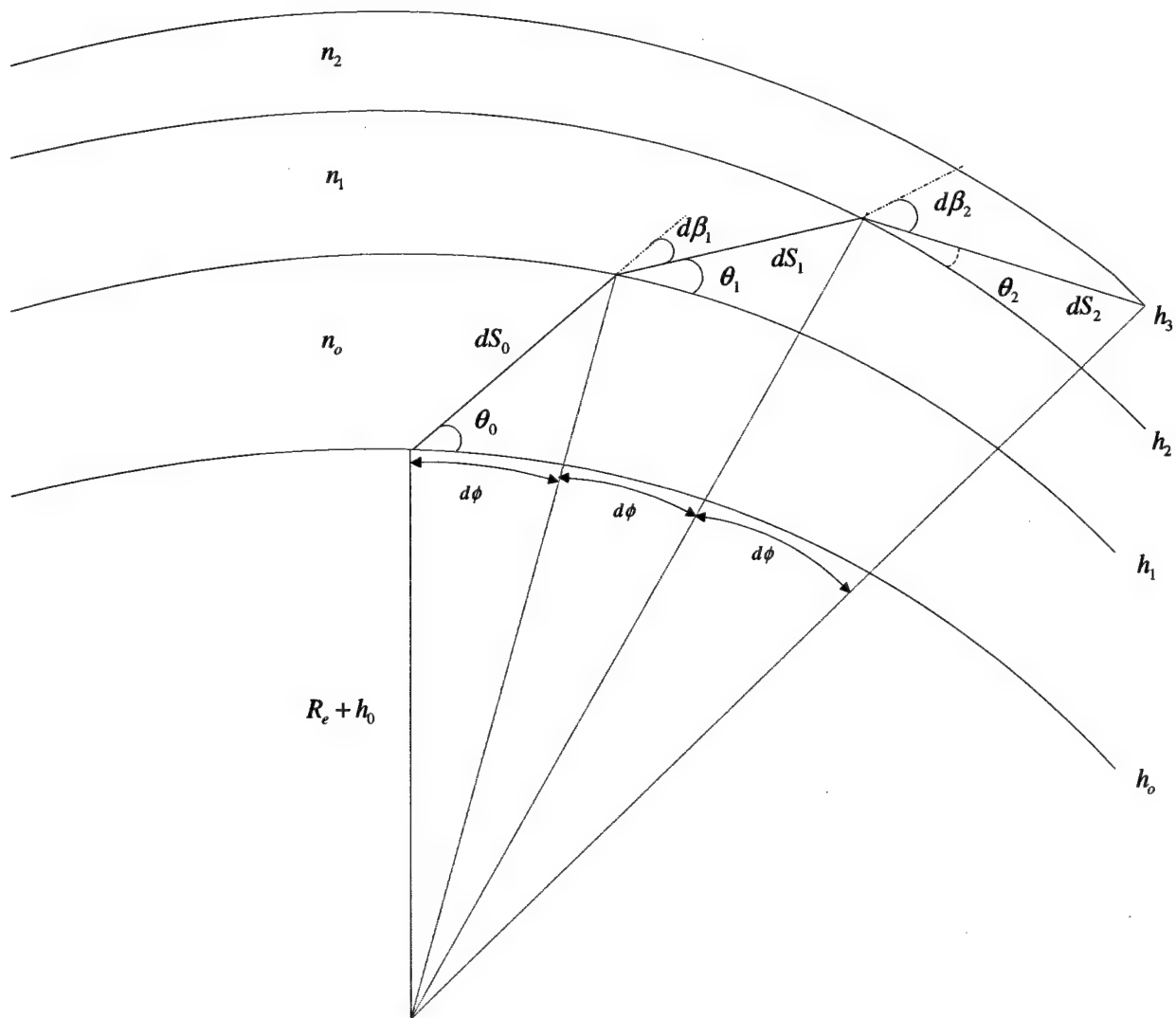
Figure 13 Round Earth Refraction Model

Also

$$d\theta_i = d\varphi - d\beta_i .$$

(30)

Combining (29) and (30) with the same manipulation as used in (19) yields

$$d\theta_i = [1 + (R_e + h_i) \ (dn_i /dh_i) \ n_i] \ d\varphi \tag{31}$$

Also, similar to (20) and (21) are:

$$dS_i = (R_e + h_i) \ d\varphi/ \cos\theta_i , \tag{32}$$

$$dSe_i = n_i \ (R_e + h_i) \ d\varphi/ \cos\theta_i , \tag{33}$$

In the limit,

$$d\theta/d\theta = 1 + (R_e + h) \ (dn /dh) /n , \tag{34}$$

$$\frac{dh}{d\phi} = (R_e + h)\tan\theta, \tag{35}$$

$$\frac{dS}{d\phi} = (R_e + h)/\cos\theta , \tag{36}$$

$$\frac{dS_e}{d\phi} = n(R_e + h)/\cos\theta = n\frac{dS}{d\phi} \tag{37}$$

$$\beta = \theta_0 + \phi - \theta \quad (or \ d\beta = d\phi - d\theta) \tag{38}$$

These equations were solved in a manner similar to that of the flat earth case resulting in

$$\phi + d\phi + d\phi + d\phi + ...$$

$$\theta = \theta_0 + d\theta_0 + d\theta_1 + d\theta_2 + ...$$

$$h = h_0 + dh_0 + dh_1 + dh_2 + ...$$

$$S = S_0 + dS_0 + dS_1 + dS_2 + ...$$

$$S_e = n_0 dS_0 + n_1 \ dS_1 + n_2 \ dS_2 + ... \tag{39}$$

These equations will constitute the truth model for evaluating any compensation algorithm. Notice that these equations, like the flat earth ones, have a singularity in the

zenith ("up") direction. Thus is due to choosing ϕ (or l) to be the independent variable, is signal propagation in the zenith direction needs to be investigated; however, this introduces a singularity in the horizontal direction, instead. The formulation given here is more useful for the Link-16 situation where low elevation signal paths are the norm. Like the flat earth case, solving these equations given the initial and final point must be done by trial and error over values of θ_0.

The true range and true elevation can be computed via

$$S_0 = \sqrt{(h_1 - h_0)^2 + 4(R_e + h_1)(R_e + h_0)\sin^2(\phi/2)}, \tag{40}$$

$$E = \arccos\left[(R_e + h_0)^2 + S_0^2 - (R_e + h_1)^2 / 2\,(R_e + h_0)S_0)\right] - 90°. \tag{41}$$

5.2.2 Time of Arrival Compensation Algorithms

We present in this section the formal derivations of two alternative TOA compensation algorithms, one based upon the "flat earth" model, and one based upon the "round earth" assumption.   The former is the basis of the existing Link-16 TOA compensation algorithm, while the latter is based on a rigorous application of ray tracing theory about an (approximately) spherical earth model.   In this discussion, it is shown that application of the flat earth model to ranges in excess of 100 km results in poor estimation of the time that the ray spends in the denser lower atmosphere resulting in unacceptable errors in TOA computations. However, it will also be shown that the flat earth model, applied to the design of the Atmospheric Filter performs extremely well when its use is restricted to relatively short ranges. The unsurprising result is that tailoring the AF design to the ranges of operation can optimize the filter's performance. This will be elaborated upon in Section 7.

5.2.2.1   Current link-16 TOA Compensation Algorithm

Figure 7 illustrates the ray propagation model used in the design of the Link-16 TOA compensation algorithm. The major assumptions are that the earth is flat and the radio ray is assumed to propagate along the straight-line path, C_0 between P_0 and P_1. Integrate

$$dt = ds/c_{air} = \frac{n}{c}ds = \frac{n}{c}\frac{dh}{\sin\theta_0} \tag{42}$$

along C_0 and call the result τ_0 ,

$$\tau_0 = \int_{c_0} dt = \int_{h_0}^{h_1}\frac{1 + ae^{-bh}}{c\sin\theta_0}dh = \frac{h_1 - h_0}{c\sin\theta_0}\left(1 - \frac{a}{b}\frac{e^{-bh_1} - e^{-bh_0}}{h_1 - h_0}\right) \tag{43}$$

Since $\dfrac{h_1 - h_0}{\sin\theta_0} = S_0,$

$$\tau_0 = \frac{S_0}{c}\left(1 - \frac{a}{b}\frac{e^{-bh_1} - e^{-bh_0}}{h_1 - h_0}\right) \qquad (44)$$

or solving for S_0

$$S_0 = \frac{c\tau_0}{1 - \dfrac{a}{b}\dfrac{e^{-bh_1} - e^{-bh_0}}{h_1 - h_0}} \approx c\tau_0\left[1 - a + \frac{ab}{2}(h_1 + h_0)\right] \qquad (45)$$
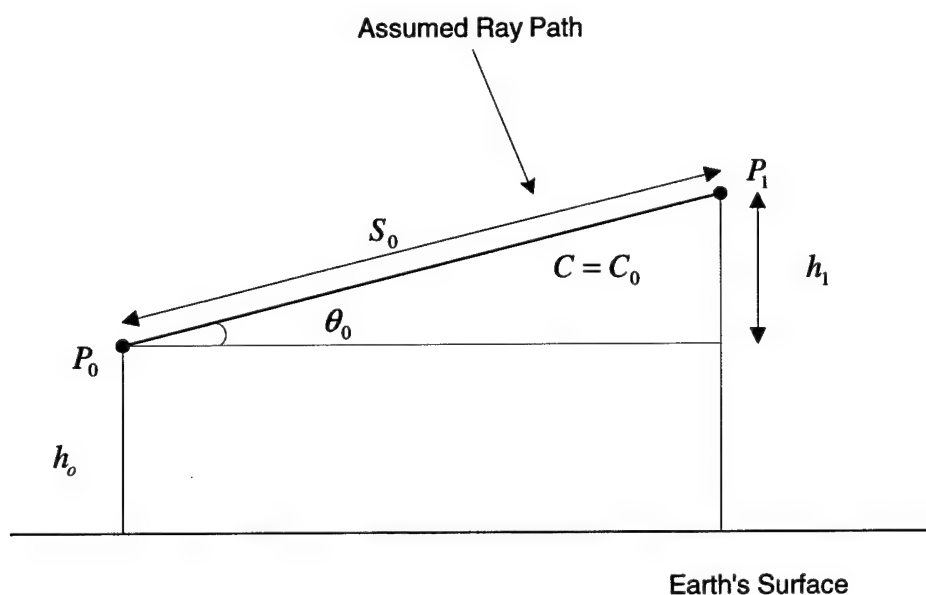
Figure 14 Link-16 TOA Compensation Model

Denote by S_{0L}, the Link-16 calculated value of true range S_0 when τ_0 is replaced by τ, the actual ray propagation time, then

$$S_{0L} \equiv c\tau\left[1 - a + \frac{ab}{2}(h_1 + h_0)\right] \qquad (46)$$

According to reference 7, the parameters a, b in (46) were picked to minimize the numerical approximations and the error arising from atmospheric variations. The resulting (fixed) values of a and b used by Link-16 are

$a = 2.95 \times 10^{-4}$,

$ab/2 = 3.625 \times 10^{-9} / \text{ft} = 1.1893 \times 10^{-8} /\text{m}$ (or $b = 0.80631 \times 10^{-4} /\text{m}$).

The current model does not perform very well at the longer ranges. For example, with parameters $a = 3.266 \times 10^{-4}$ and $b = 1.3061 \times 10^{-4} /\text{m}$ and with $S = 400$ km, the implemented Link-16 compensation is in error by 22.7 m and the so-called exact compensation is in error by 20.6 m. The explanation is as follows: Consider the case $\theta_0 = 0$, $h_0 = 0$ with $a = 3.266 \times 10^{-4}$, $b = 1.3061 \times 10^{-4} /\text{m}$. When $S = 400$ km, the true altitude of the ray is 9680m. The link-16 compensation algorithm implicitly is using an initial ray elevation angle of arcsin (9.680/400) = 1.39°. Although not much different from the true value zero, it vastly changes the ray's altitude profile, with the result of having it spend much too small an amount of time in the denser lower atmosphere. For example, after the ray has traveled 100 km, its actual altitude is 573 m compared with the Link-16 model's implicit assumption of 2500 m.

5.2.2.2 An Improved TOA Compensation Algorithm

It is possible to do much better, consider Figure 15. It is not difficult to show that

$$h_1 \approx h_0 + \theta_0 R_e \phi + \frac{R_e}{2} \phi^2 = h_0 + \theta_0 l_e + \frac{1}{2R_e} l_e^2 . \tag{48}$$

Since in Link-16 applications both h_0 and h_1 are known (see discussion at the beginning of this section) and ϕ can be computed from other known information, or computed to sufficient accuracy using $\phi \approx S_e / R_e$; θ_0 in (56) can be solved for. Then (44) can be used to approximately describe the ray path as a function of ϕ. Also, from (36) and (37)

$$\frac{dL_e}{d\phi} = \frac{d(S_e - S)}{d\phi} = (R_e + h)\frac{n-1}{\cos\theta} = R_e(n-1) \tag{49}$$

in situations where $\cos\theta \approx 1$. Integrating (49) produces

$$L_e \approx R_e \int_0^\phi (n-1)d\phi . \tag{50}$$

Using the exponential model (4) to model n, the excess path length can be approximately computed as

$$L_e \approx S_e - S \approx R_e \int_0^\phi (n-1)d\phi = R_e \int_0^\phi \exp(-bh)d\phi$$

$$\approx a R_e \int_0^\phi \exp\left[-b\left(h_0 + \theta_0 R_e \phi + \frac{R_e}{2}\phi^2\right)\right]d\phi \tag{51}$$

Discussion of the evaluation of this integral will be deferred until after a slightly more general form is derived.
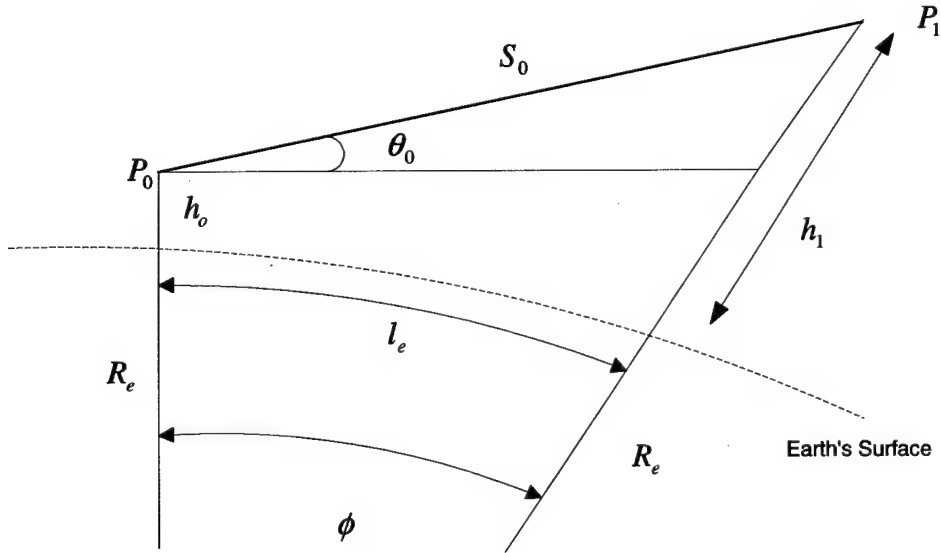
Figure 15 Altitude Variation

Considering (34), notice that the term on the right hand side is 1/k. with k as defined in (3). Thus (34) can be written as

$$\frac{d\theta}{d\phi} = 1/k. \tag{52}$$

Now assume k is constant, then integration of (52) yields

$$\theta = \theta_0 + \phi/k. \tag{53}$$

Then from (35) and (53)

$$\frac{dh}{d\phi} = (R_e + h)\tan\theta \approx R_e \, \theta \approx R_e \, \theta_0 + \frac{R_e}{k}\phi. \tag{54}$$

Integrating (48) yields

$$h \approx h_o + \theta_0 \, R_e \, \phi + \frac{R_e}{2k}\phi^2 \approx h_o + \theta_0 l_e + \frac{1}{2kR_e} l_e^2 \tag{55}$$

which is identical to (48) except for the incorporation of k. The second form of (48) makes it clear why k is referred to as the effective earth radius factor. Using (55) instead of (48) in (50) produces

$$
\begin{aligned}
L_e &\approx a R_e \int_0^\phi \exp\left[-b(h_0 + \theta_0 R_e \phi + \frac{R_e}{2k}\phi^2) \right] d\phi \\
&= a R_e \exp(-bh_0) \int_0^\phi \exp\left[-\frac{bR_e}{2k}(\phi^2 + 2k\theta_0\phi) \right] d\phi \\
&= a R_e \exp(-bh_0)\exp(\frac{b\theta_0^2 kR_e}{2}) \int_0^\phi \exp\left[-\frac{bR_e}{2k}(\phi + k\theta_0)^2 \right] d\phi \\
&= a R_e \exp(-bh_0)\exp(\frac{b\theta_0^2 kR_e}{2}) \frac{1}{\sqrt{\frac{bR_e}{2k}}} \int_{\sqrt{\frac{bR_e}{2k}}(k\theta_0)}^{\sqrt{\frac{bR_e}{2k}}(\phi+k\theta_0)} \exp\left[-\frac{bR_e}{2k}(\phi + k\theta_0)^2 \right] d\phi \\
&= \frac{\sqrt{\pi}}{2} \frac{a R_e \exp(b(\theta_0^2 kR_e/2 - h_0))}{\sqrt{\frac{bR_e}{2k}}} \times \left[erf\left(\sqrt{\frac{bR_e}{2k}}(\phi + k\theta_0) \right) - erf\left(\sqrt{\frac{bR_e}{2k}}(k\theta_0) \right) \right] \\
&= \frac{\sqrt{\pi}}{2} \frac{a R_e \exp(b(\theta_0^2 kR_e/2 - h_0))}{\sqrt{\frac{bR_e}{2k}}} \times \left[erfc\left(\sqrt{\frac{bR_e}{2k}}(k\theta_0) \right) - erfc\left(\sqrt{\frac{bR_e}{2k}}(\phi + k\theta_0) \right) \right]
\end{aligned}
$$

(56)

where

$$
erf(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x \exp(-u^2)\, du \tag{57a}
$$

and

$$
erfc(x) \equiv 1 - erf(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \exp(-u^2)\, du \tag{57b}
$$

are the error function and complementary error function, respectively. Simple and fast numerical routines exist to evaluate these integrals in much the same manner as a sine or cosine.

Also from (34) and (38) and again assuming (4),

$$\frac{d\beta}{d\phi} = -(R_e + h)\frac{dn/dh}{n} = (R_e + h)\frac{b(n-1)}{n} \approx R_e b(n-1). \tag{58}$$

Integrating (58) gives

$$\beta \approx bR_e \int_0^\phi (n-1)\,d\phi \tag{59}$$

and thus

$$\beta \approx bL_e . \tag{60}$$

Then from (53)

$$k \approx \frac{\phi}{\theta - \theta_0} = \frac{\phi}{\phi - \beta} = \frac{1}{1 - \frac{\beta}{\phi}} \approx \frac{1}{1 - \frac{bL_e R_e}{S_e}} \quad . \tag{61}$$

Using the above, the algorithm to approximately compute L_e is to first start with $k = 1$ and solve (55) for θ_0 as

$$\theta_0 = \left(h_1 - h_0 - \frac{S_e^2}{2kR_e} \right) \Big/ S_e \tag{62}$$

Then compute a first estimate of L_e using (56). Next, using this L_e, compute k using (61) and then repeat (62) and (56). This can be repeated several times. This technique works extremely well, but before discussing the results, a method to compensate the curvature correction will be L_c derived.

Consider Figure 9, which shows the arc of a circle with central angle θ. The arc has length S and the corresponding chord as length S_0.
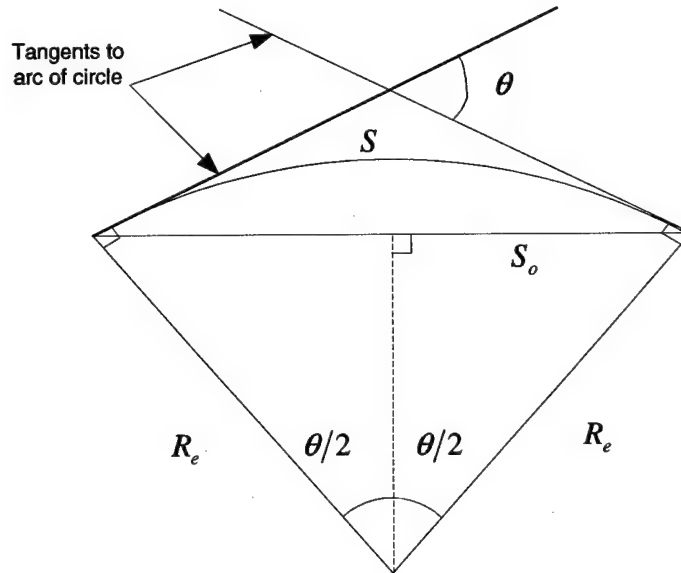
Figure 16 Arc to Chord Correction

The difference in lengths is

$$S - S_0 = R\theta - R\sin(\theta/2) \approx \frac{R\theta^3}{24} = \frac{R(S/R)^3}{24} = \frac{S^3}{24R^2} \approx \frac{S_0^3}{24R^2} \tag{63}$$

for small θ. The ray path C in Figure 1 will be approximated as the arc of a circle with length S and central angle β resulting in a radius of curvature of $R_r \approx S/\beta \approx S_e / \beta$. Then

$$L_C = S - S_0 \approx \frac{S_e^3}{24R_r^2} \approx \frac{S_e^3}{24(S_e/\beta)^2} = \frac{S_e\beta^2}{24} \approx \frac{S_e(bL_e)^2}{24} \tag{64}$$

Thus, the curvature correction can be approximated using excess path length.

Table 4 tabulates some results of the proposed compensation method. As can be seen, the method works very well, the maximum error in the nominal case being ±1 meter. [REMOVE?? Comparing Table 4 with Table 7(d)??? shows that the error function based algorithm performs within a meter or two of perfection, i.e., any error in Table 4 is strictly due to atmospheric variation (the sign differences are irrelevant).]
The proposed algorithm was also evaluated with refraction calculated using the UNB1 atmospheric model – the errors were typically less than one meter with a maximum error of less than three meters.

Error Function based Estimate of S – S minus actual S – S (m)																
h_{rcvr} (km)	h_{emit} (km)	S = 150 km (81 nm)			S = 250 km (135 nm)			S = 350 km (189 nm)			S = 450 km (243 nm)			S = 550 km (297 nm)		
0	5.2	8	0	-7	15	0	-13									
	10	5	0	5	9	0	-9	16	0	-14						
	15	3	-1	-4	6	0	-6	10	0	-10	17	0	-15			
5.2	5.2	4	0	-4	7	0	-8	13	0	-12	20	0	-18	33	1	-26
	10	2	0	-3	4	0	-5	7	0	-8	12	0	-12	20	0	-18
	15	1	0	-2	2	0	-3	3	0	-5	6	0	8	11	0	-12
10	10	0	0	-1	1	0	-3	2	0	-4	5	0	-7	10	0	-11
	15	0	0	-1	0	0	-2	0	0	-3	1	0	-4	4	0	-7
15	15	-1	0	0	-1	0	-1	-2	0	-1	-1	0	-2	0	0	-4

Note 1: Blank entries indicate solution did not exist.

Note 2: The first value in each triplet is for a = 240 x10^{-6} and b = 0.01017/km , the second value is for a = 326.6 x10^{-6} and b = 0.13061/km and the third value is for a = 400 x10^{-6} and b = 0.1411/km.

Note 3: Since the situation is symmetrical with respect to h_0 and h_1, redundant results are not tabulated.

Table 4 Error Function Based TOA Compensation Error versus h_0 , h_1 and S .

5.2.3 Working Approximations to Round Earth Excess Path Length

Equation (56) for the excess path length, L_e in the round earth model is correct, but is not yet in a form suitable for application to Kalman Filter design. We therefore must seek a simpler, yet accurate approximation to this complex expression in order to proceed further.

We begin our derivation with a simplified version of Equation (56), for L_e, with K set to unity. Setting K to unity will be shown to have only second order effects on the final Kalman estimation algorithm.

We have

$$L_e = aR_e \int_0^{\phi} \exp\left[-b\theta_0 R_e \phi - \frac{bR_e}{2}\phi^2\right] d\phi \qquad (65)$$

We make the following substitutions.

Let:

$$C = e^{-bh_0}\, a R_e,\ \beta = {b\theta_0 R_e}\big/{2},\ \alpha = {bR_e}\big/{2}. \tag{65a}$$

Then:

$$L_e = C \int_0^\phi \exp\!\left[-2b\phi - \alpha\,\phi^2\right] d\phi \tag{66}$$

This expression may be integrated in closed form to yield:

$$
\begin{aligned}
L_e &= C\left[\frac{1}{2}\sqrt{\frac{\pi}{\alpha}}\, e^{-\beta^2/\alpha}\, erf\!\left(\sqrt{\alpha}\; x + \frac{\beta}{\sqrt{\alpha}}\right)\right]_0^\phi \\
&= C\frac{1}{2}\sqrt{\frac{\pi}{\alpha}}\, e^{-\alpha\theta_0^2}\left[erf\!\left(\sqrt{\alpha}(\phi+\theta_0)\right) - erf\!\left(\sqrt{\alpha}\theta_0\right)\right]
\end{aligned}
\tag{67}
$$

This expression is of the form:

$$L_e = C\, e^{-\alpha\theta_0^2}\left\{g(\phi+\theta_0) - g(\theta_0)\right\} = C\, e^{-\alpha\theta_0^2}\left[\Delta g(x)\right] \tag{68}$$

where:

$$g(x) \approx e^{-\alpha x^2}\left[x + \frac{2}{3}\alpha x^3\right] \tag{69}$$

and we have replaced the erf(x) function with the first two terms of its series expansion.

Evaluating the derivative of g(x) at $x = \theta_0$:

$$\left.\frac{dg(x)}{dx}\right|_{x=0} = e^{-\alpha\theta_0}\left[1 - \frac{4}{3}\alpha^2\,\theta_0^4\right] \tag{70}$$

Then:

$$\Delta g(x) \cong e^{-\alpha\theta_0^2}\left\{1 - \frac{4}{3}\alpha^2\,\theta_0^2\right\}\phi \tag{71}$$

The final approximation for L_e is then:

$$L_e \cong C e^{-2\alpha\theta_0} \phi \qquad (72)$$

with C as given by (65a).

5.3 LINK-16 Navigation Overview

Before we proceed further, it is useful to give a short introduction to the Link-16 Tactical Data Link System.

The basis of Link-16 navigation is a Hybrid Inertial Solution combining Inertial Navigation System (INS) acceleration measurements with external range measurements provided by Precise Position Location and Identification (PPLI) messages, and, if available, GPS position and time observations. These external measurements are processed on each individual platform within a 20-state Extended Kalman Filter, designated as the L16 Navigation Kalman Filter.

BAE SYSTEMS

Hybrid Navigation Diagram
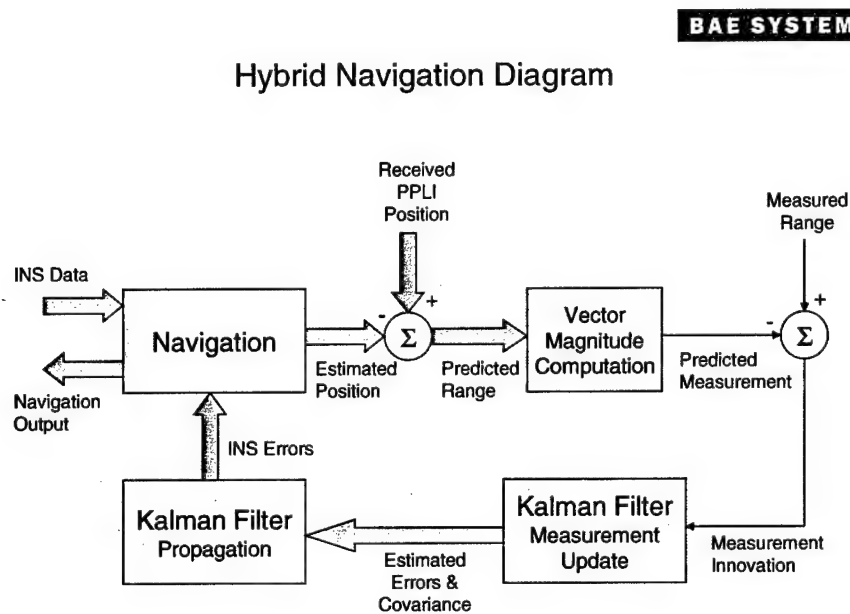
Figure 17 Hybrid Navigation/ Kalman Filter Flow

Link 16 Navigation Kalman Filter State Configuration

•The software to perform the navigation function is based on a 20-state Kalman filter.

•The TOA/PPLIs and GPS measurements are used as observations in the Kalman filter, and the host velocity information from the INS is integrated with the observations for propagating position.

STATE ELEMENT	SYMBOL	DESCRIPTION
1	$\varepsilon\lambda$ †	Latitude position error
2	$\varepsilon\phi$ †	Longitude position error
3	εh	Altitude error
4	εv_X	(X or North) local level velocity error
5	εv_Y	(Y or West) local level velocity error
6	εkh	Altitude scale factor error
7	$\varepsilon\theta_x$ ††	(X or North) local level misalignment
8	$\varepsilon\theta_y$ ††	(Y or West) local level misalignment
9	$\varepsilon\theta_z$ ††	(Z or Azimuth) local level misalignment
10	εv_{BX} ††	(X or North) damping velocity bias error
11	εv_{bY} ††	(Y or West) damping velocity bias error
12	εP_U	U-axis relative grid position error
13	εP_V	V-axis relative grid position error
14	$\varepsilon\beta$	Relative grid azimuth error
15	εv_{U_c}	Relative grid drift U-axis velocity error
16	εv_{v_c}	Relative grid drift V-axis velocity error
17	εB_C	Synchronization clock bias error
18	εf_C	Synchronization frequency bias error
19	εf_D	Synchronization dynamic frequency error
20	εf_{DR}	Synchronization dynamic frequency rate error

2

Figure 18 Hybrid Navigation/ Kalman Filter States

An extremely important feature of Link-16 navigation is that it provides to an entire participating community accurate correlation in position, velocity, and time in both absolute (i.e. WGS-84) geodetic coordinates, and relative coordinates, performed simultaneously. While the WGS-84 solution is always referenced to the earth's geoid, the relative solution is centered at an arbitrary location on the earth's surface known to all participants.

Intimately related to Link-16 navigation performance is the precise time synchronization achieved throughout the community, with under ten nanoseconds a representative figure. This precision is sought for one reason only – the need to accurately convert measured time of arrival of a message to that of a pseudorange measurement needed for navigation. This synchronization process progresses through three stages, namely Initial Entry, Coarse Synchronization, and Fine Synchronization. Initial Entry is the process of receiving a single message identifying the specific TDMA slot corresponding to the community time base. Coarse Synchronization requires a second confirming message, while Fine Synchronization

refines time uncertainty from a few milliseconds to the desired nanosecond level. Fine Synchronization also involves the calibration of the local oscillator time and frequency to community standards. Fine Synchronization can be achieved via active (i.e. RTT Message), or passive (as part of navigation solution).

5.3.1 Link-16 Navigation Community Measurement Protocols

The PPLI message is the basic medium for community navigation data transfer within the Link-16 community. It contains, among many other variables, the estimated geodetic and relative position, course, speed, and qualities (needed for proper Kalman weighting by data recipients). It should be noted that the use of the PPLI range measurements by a recipient of the message does not constitute a "triangulation" process. Rather, the received data is optimally combined with the inertial acceleration measurements to generate a true optimal solution based upon all available information.

There exists a hierarchy of roles within the navigation community in order to assure the unidirectional flow of information from platforms of higher quality to those of lower quality. At the highest level is the Navigation Controller (NC), which in the C-17 scenario will be assigned to the formation leader. In the relative grid, the navigation controller is assumed to have perfect positional accuracy, and thus is assigned a relative position Quality (Qpr) equal to 15. Under some circumstances, there may also be assigned a Secondary Navigation Controller, but this option will not apply to our scenario. The second level is that of Primary User (PU), which will be assigned to formation subleaders. Primary Users will have high relative navigation qualities, but not equal to the Navigation Controller. At the lowest level are the individual wingmen designated as Secondary Users, with still lower quality levels. In all cases, relative grid information will flow from platforms of higher quality to those of lower quality, under the control of the Navigation Kalman Filter and its observation screening algorithms. The Kalman filter will automatically extract optimal sets of both relative and absolute geodetic navigation from each PPLI simultaneously, in order to maintain the best possible navigation solution in both coordinate frames. Figure 3 indicates the flow patterns of information for relative navigation, absolute navigation, and time synchronization.

Navigation Community Hierarchy
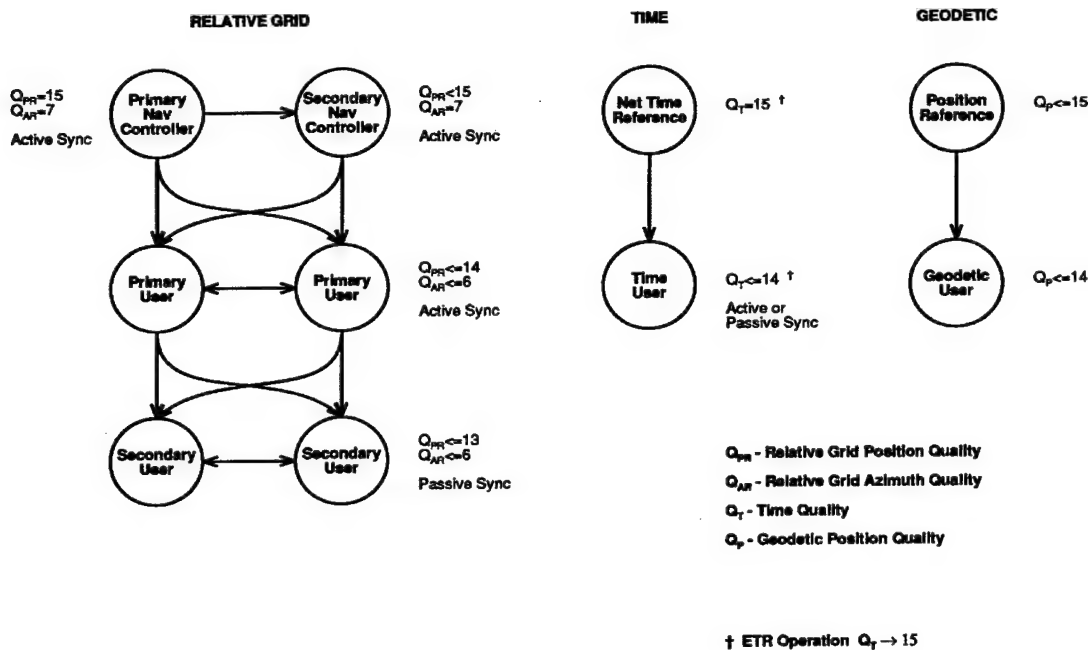
Figure 19 Navigation Community Hierarchy

The relationship of the relative and geodetic solutions will depend on the availability of GPS data anywhere within the community. If even a single platform has access to accurate geodetic data via GPS or other source, the community relative and geodetic solutions will tend to coalesce. This transfer of geodetic information throughout the community takes place via the PPLI messages themselves.

The hybrid integration of the navigation solution with the on-board inertial measurement device means that a relatively low update rate from potential sources is required to maintain accurate navigation; normally, PPLIs would be transmitted at a nominal rate of once per 12 seconds per platform. A source screening process is applied at each platform to accept the optimum subset of potential measurements which can aid the absolute and relative solutions. The source selection processing will select this optimal group based on a complex function of reported qualities, positional geometry, and the covariance of the hybrid solution itself (Figure 18).

BAE SYSTEMS

Source Selection Function Diagram
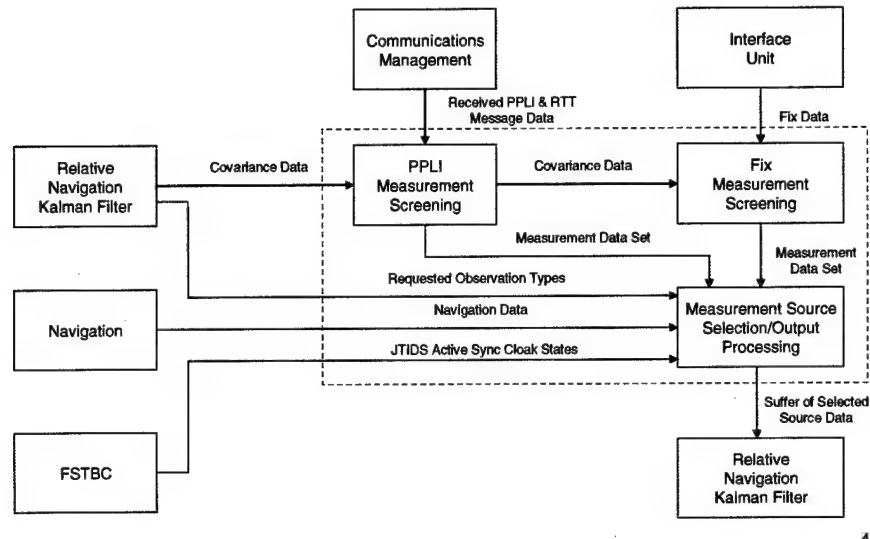
Figure 20 Link-16 Navigation Source Selection

5.3.2 Link-16 Navigation Performance

Link-16 navigation is now an operational reality in many US triservice, NATO, and other foreign aircraft, ships, and ground installations. It is a mature technology, with more than 34 years of continuous development and refinement. The performance parameters discussed in this section are the results of a sophisticated simulation package using actual operational flight software, and have been verified via precision tracking and recording measurements during many thousands of hours of testing and operational use.

Figure 21 indicates the data flow of the Link-16 navigation algorithms. Please note that the model faithfully represents the host navigation interface necessary for integration of INS inputs to the algorithms.

Terminal Navigation Function Diagram
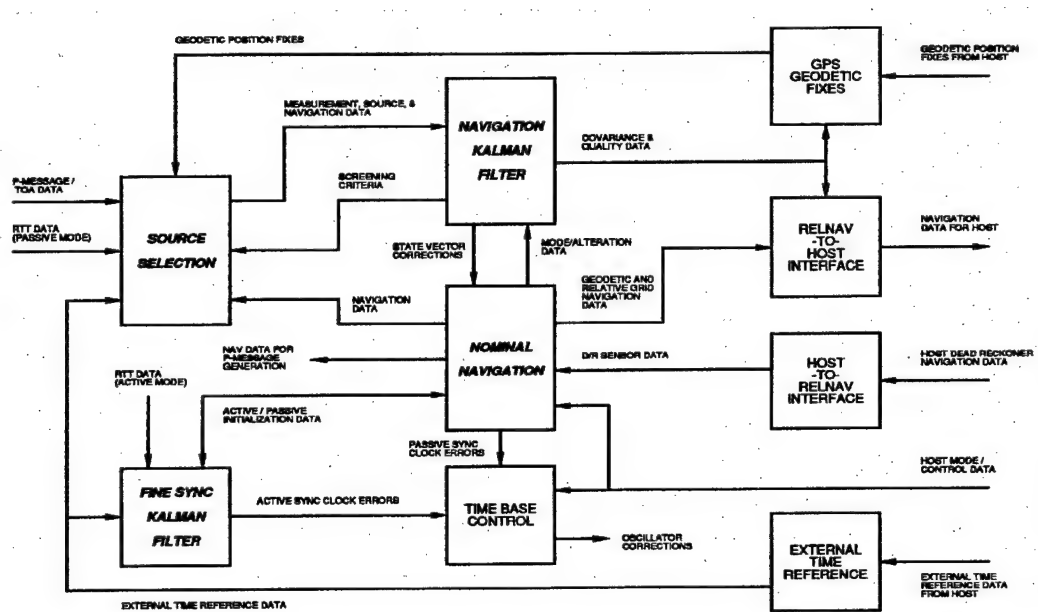
Figure 21 Link-16 Navigation Functional Flow

The results obtained by the current Link 16 system, although impressive, can still be improved upon. One approach, described in this report, is to augment existing performance with the Atmospheric Filter (AF), which is designed to refine the range estimates of the current Link-16 navigation solution. The AF results would be input to the Link-16 Navigation system via the Source Selection module shown in Figure 18.

6   METHODS, ASSUMPTIONS AND PROCEDURES

6.1.1 Simulation Overview

The Atmospheric Filter (AF) algorithm designs described above were simulated as Kalman Filters in a FORTRAN program. As the AF measurement model is nonlinear, the Extended Kalman Filter (EKF) formulation was used. EKFs were implemented for the both the Flat Earth (FE) and Round Earth (RE) versions of the AF and have state vectors based on the Exponential model of atmospheric refractivity. The state vector elements were functions of the parameters, a and b, of the exponential refraction model. The RE utilizes the additional state, k, the ray curvature parameter.

In the FORTRAN program, a truth model using the more accurate RE model is used to calculate the real world range value, S_0. The range obtained from the state estimate is differenced from S_0 to create the range errors. The range estimate for FE is equation (46) and that of RE is c * TOA minus the approximation to L_e, equation (72).

This section describes this implementation and exhibits the results for a range of operational conditions simulated in the test cases.

6.2 Atmospheric Refraction Simulation

The AF Extended Kalman Filters were simulated using a FORTRAN program. The state vectors for both the Flat Earth (FE) and Round Earth (RE) algorithms were based on the Exponential model of atmospheric refractivity. The state vector elements for FE were the parameters, a and b, themselves. Those of the RE are

$$\alpha = \left(\frac{b}{2}\right) R_E, \quad \gamma = a R_E, k = \text{curvature} \text{ , where } R_E \text{ is the Radius of the Earth}$$

(approximately 6378 km).   (73)

Besides the Exponential model, two other two models of atmospheric refractivity were considered. They depend upon meteorological data such as temperature and pressure but it has been shown in a previous section (cf. Figures 3 through 10) than the Exponential model is a very good fit to their results. So, a Kalman Filter having a state vector based on the Exponential model can be used for these models, too. Furthermore, as the Smith-Weintraub model has been shown to give virtually identical results to the third candidate (Thayer) model only its results will be compared to those of the exponential model.

The FORTRAN program's truth model generates pseudo measurements of the Time of Arrival (TOA) from which the real world refracted signal path length, S_0, is derived. Subtracting these from the AF estimates yields the AF range errors.

In the simulation, there is one Link-16 unit ("the receiver") receiving TOA information from three other Link-16 platforms (the "emitters"), each transmitting PPLI messages to it every three seconds. Thus, the receiver receives one such message from each of the other units every second in a round-robin fashion. The program simulates 90 seconds of PPLI and real time processing of the EKF. The EKF estimates one state each for all three emitters.

Refractivity in the atmosphere is approximately the same in a fairly large volume of the atmosphere. At any given time, one would expect its parameters to be fairly constant all throughout an area of the earth in which the weather is the same. It is also expected to vary much more slowly in time than Link-16 unit positions would. Operationally, it is natural to assume that one refractivity state would suffice for all emitters which are in one particular volume of the atmosphere. At any given time, therefore, one would expect that only a few AF filters would have to be running simultaneously for a given Link-16 receiver. A characteristic of Kalman filters is that the state estimation will improve as a function of time as long as measurements are being processed. *This means that an ensemble of Link-16*

range estimates will help create refractivity estimates which will lead to higher quality range estimates for each platform due to a phenomenon similar to that of sensor registration, which is described above.

For simplicity, each of the Link 16 units in the simulation is stationary. This clarifies range estimation issues that depend only on different altitudes and ranges without the additional complexity of dealing with emitter motion. In a later phase of the study, realistically mobile models of participating platforms will be obtained from the BAE Link-16 Simulation software to better evaluate the system.

The factor k[16], which changes the effective earth radius in the RE model, has an effect on the error and is a natural RE model parameter. For a given geometry of LINK-16 emitter and receiver, varying k was found to improve the results. The best way to estimate k is to make it another Kalman filter state.

Snell's Law for a ray of light going through a *plane* surface boundary from a medium of one particular constant refractivity value to another depends on the *sine* of the angle between the ray and the angle to the normal to the plane. If the layers of the medium are *curved* as in the layers of the atmosphere, then the refraction can be shown[17] to vary as the *cosine* of the angle to the tangent of the circular boundary and *also* to the radius of curvature of the boundary. For small angles, it can be shown that Snell's law involves a term k times R_e, where R_e equals the Earth's radius, so k is taken to be a curvature factor.

It was seen that if k were a certain value greater than 1 for a given situation, then the curvature of the ray path could be reduced to the point in which it was nearly zero, a straight path[18]. This greatly simplified calculations in the graphical way they were performed before computers were powerful enough to solve problems such as this one. Here, however, k is useful as another degree of freedom in helping the EKF arrive at an estimate; k will help warp the world for the EKF to facilitate a solution. In the case of constant refractivity everywhere, the implemented EKF estimates k to be approximately 1. This is as expected, so the three state EKF used for the RE model has the right asymptotic behavior.

$k * R_e$ equals the effective Earth radius: this is the radius of a hypothetical Earth[19] for which the distance to the radio horizon, assuming rectilinear propagation, is the same as that for the actual Earth with an assumed uniform vertical gradient of atmospheric refractive index. *Note:* For the standard atmosphere, the effective Earth radius is 4/3 that of the actual Earth radius. The radio horizon is the locus of points at which direct rays from an antenna are tangential to the surface of the Earth. In practice, k can vary from about 0.5 to about ·4.0.

---

[16] Reed & Russell, Reference 5, pp. 37-40.
[17] Ibid., pp. 43-45.
[18] Ibid., p. 38.
[15] Ibid., pp. 43-45.

6.2.1 Extended Kalman Filter Basics

Before the Atmospheric filter is described, the Kalman Filter itself will be reviewed, so it can be understood by a reader unfamiliar with it.

Kalman filters are a common tool to calculate best estimates of a system state if only noisy sensor data are available. It is used in a variety of inertial navigation systems and also in tracking devices[20]. Such filters consist of a predictor and a corrector. The latter uses sensor information, while the former – generally speaking – uses a model of the underlying dynamics. Problems arise if the model does not describe the dynamics properly.

Consider a nonlinear system:

$$\dot{x} = f(x,t) + w, \quad w \sim N(0,Q),$$
$$y = h(x,t) + u, \quad u \sim N(0,R). \tag{74}$$

The function f models the dynamics of the state vector $x(t)$. Since f is used to estimate the state of the system, it is referred to as the predictor. The function h is the model of measurements y. It is used to correct the prediction and referred to as the corrector. w and u are the zero-mean Gaussian white system and measurement noise. The Extended Kalman Filter (EKF) calculates $\hat{x}$, the best estimation of the state x, and its covariance P:

$$\dot{\hat{x}} = f(\hat{x}) + PH^T R^{-1}(y - h(\hat{x})),$$
$$\dot{P} = FP + PF^T + Q - PH^T R^{-1}HP, \tag{75}$$

with the local linearizations $F = \left.\dfrac{\partial f(x)}{\partial x}\right|_{x=\hat{x},t}$ and $H = \left.\dfrac{\partial h(x)}{\partial x}\right|_{x=\hat{x},t}$.

F is taken to be the unit matrix as we model the states as random constants (or biases).

The standard application is to integrate the dynamic model into f. The dynamic model is used to calculate the prediction of the next state $\tilde{x}_+$. The measurements $y(x)$ are used to correct this prediction.

The basic state vector, x, is

$$State: \quad x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{76}$$

In it, a and b are the parameters of the exponential refractivity model and are treated as random constants in the Kalman Filter model state vector.

---

[20] Gelb, A., ed., *Applied Optimal Estimation*, The Analytical Sciences Corp., MIT Press, 1974.

6.2.2 Atmospheric Filter Kalman Filter Models, Basic Equations

The two Atmospheric Filter EKF models considered in this study differ in how the atmosphere is modeled:, in one, the Flat Earth, or FE model, the earth is considered to be flat, in the other, the ray Tracing or Round Earth (RE) model, a more realistic but more complex model, the earth is considered to be perfectly round and signals propagating in it are modeled as rays traced through spherical layers of the atmosphere.

Common to both models are the following terms (refer to Figure 15 for ϕ):

KC = Speed of light in vacuo, that is, c, quantized in LINK-16 time units or counts.

ϕ = The interior angle between lines measured from the center of the earth to the emitter and receiver, respectively.

Measurement, Y = TOA   (77)

The states of both the FE and RE EKFs are derived from the Exponential refractivity model. In both models, the range error depends upon the explicit form of the index of refraction. The exponential form of the index of refraction has been shown to be realistic in section 6.1.4, above. This is crucial because the results of the Kalman Filter depend on its having a realistic state model. The measurements are the times of arrival of the Link-16 PPLI signals; the refraction is not measured directly. Therefore, there must be a transformation from the mathematical measurement space to refractivity state space. A fairly large region of the atmosphere can be characterized by one more or less constant value of the one set of values of the refractivity parameters. In such a volume of the atmosphere, a number of Link-16 units could emit from a range of positions. The signals from all these emitters will then experience refractivities characterized by the same values of the parameters a and b even though the emitters are dispersed geographically. Each signal will then contribute to improving the state estimate.

The FE EKF has state elements a and b, the parameters of the exponential refractivity, and the RE EKF has two state elements which are functions of a and b, namely,

$$\gamma = a * R_e \text{ and } \alpha = b * R_e / 2,$$ (78)

where R_e = the Earth's radius, and a third state, k, the coefficient of earth curvature.

In practice, the value of k can vary in value from 0.5 to about 5.0. Its average value in temperate climates is about 1.33 (see References 2 and 3). Runs made with the index of refraction held constant throughout (that is, no refraction) had the value of k come out to be 1, as expected, so the filter gives the right result in the trivial case. The state vector element k gives the RE EKF another degree of freedom to better estimate the path error.

6.2.2.1 FE Model Specifics

$$\text{State}: X_{FE} = \begin{bmatrix} a \\ b \end{bmatrix}$$

(79)

Where the states a and b are taken from the index of refraction $n = 1 + a * e^{-bh}$ (Exponential Model)

$$Q_{FE} = \begin{bmatrix} 10^{-6} & 0 \\ 0 & 10^{-6} \end{bmatrix} \quad \text{(nominal values)}$$

(80)

$$R_{FE} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{(nominal values)} \tag{81}$$

$$P_{FE}(\text{initial}) = \begin{bmatrix} \sigma_{a0} & 0 \\ 0 & \sigma_{b0} \end{bmatrix} \tag{82}$$

where the nominal values $\sigma_{a0} = 461.*10^{-7}$, $\sigma_{b0} = 1.13*10^{-5}$

$$H_{FE} = \left[(S_0/KC) * FAC1/FAC2**2 \quad -(S_0/2.*KC)*(XF(1)*h_0 + h_1))/FAC2**2 \right] \tag{83}$$

Where FAC1 = $(1.0 - 0.5 * X_{FE}(2) * (h_1 + h_0))$ and
FAC2 = $(1.0 - X_{FE}(1) + 0.5 * X_{FE}(1)*X_{FE}(2) * (h_0 + h_1))$

$$S_0^{Estimate} = Y * KC * FAC2 \tag{84}$$

<u>FE Model TOA:</u>

$$\text{TOA (12.5 nsec counts)} = S_0 / (K_C * (1 - a + 0.5 * a * b * (h_{emitter} + h_{receiver}))) \tag{85}$$

6.2.2.2 RE Model Specifics

$$\text{State}: X_{RE} = \begin{bmatrix} \gamma \\ \alpha \\ k \end{bmatrix}, \text{ where } \alpha = \left(\frac{b}{2} \right) R_E, \ \gamma = aR_E, k = \text{curvature}, R_E = \text{earth radius}$$

(86)

$$P_{RE}(\text{initial}) = \begin{bmatrix} \sigma_{a0} & 0 & 0 \\ 0 & \sigma_{b0} & 0 \\ 0 & 0 & \sigma_{k0} \end{bmatrix} \quad \text{where initial values } \sigma_{a0} = 2940, \ \sigma_{b0} = 3600, \ \sigma_{k0} = 2$$

(87)

Where FAC $= -2.* (\theta_0^2 + (h_0/R_{Earth}))$,
FAC1 $= FAC * X_{RE}(2)$, and
$L_e = X_{RE}(3) * X_{RE}(1) * exp(FAC1) * \phi$

$$R_{RE} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{(nominal values)}$$

(88)

(89)

$$S_0^{Estimate} = Y * KC * L_e$$

RE TAU (TOA):
$$H_{RE} = \begin{bmatrix} L_e/(KC * X_{RE}(1)) & FAC * L_e/K & L_e/(KC * X_{RE}(3)) \end{bmatrix}$$

(90)

$$TOA = \tau \ (12.5 \ nsec \ counts) = (L_e + L_C + S_0)/ \ K_C$$

(91)

6.3 SAMPLE CASES: RESULTS AND DISCUSSION

In the simulation results presented below, one Link-16 unit (called "the receiver" or "home") receives TOA information from three other Link-16 platforms each transmitting PPLI messages to it every three seconds. Thus, the home unit is receiving one such message from each of the other units every second in a round-robin fashion.

Each scenario involves three emitters; the results are shown for the range to one of them as estimated at the receiving unit.

The table entries show results for the magnitude of the range error using both the current Link-16 model and the Atmospheric Extended Kalman Filter (AF) using the FE and RE models of the physical atmosphere. All the AF results shown use the exponential atmospheric refractivity model. The results using the Smith-Weintraub refractivity model are very similar.

The results are summarized in Table 6. In it, the FE and RE results are shown along with those of the currently implemented Link-16 compensation model. (These are taken from Reference 4, Table 9, page 23.)

As can be seen from the table, the FE AF is generally better than the RE AF in estimating ranges at true ranges of less than 250 km between the emitter and receiver. The three state RE EKF does much better at true ranges of 250 km or more. Both AF models generally do better than the current Link-16 compensation algorithm at true ranges of 250 km or more.

At true ranges of less than 250 km, the FE routine generally does better than the current Link-16 model, but the RE does not, except at higher emitter-receiver elevation angles. As the Link-16 model uses average values, it is expected to work well some of the time; it seems to be optimized for short range estimation at lower elevation angles. Operationally, the quality of the AF output would be monitored and the compensation routine in the Link-16 Operational Computer program (OCP) could be set to default to the current model if the AF output quality were below a certain value in the short range case.

The current Link-16 compensation model for the range S estimation is:

$$S_{L16} = c * TOA * [1 - a + ab/2 \, (h_{emit} + h_{rcvr})], \qquad (92)$$

where $a = 2.95 * 10^{-4}$ and $ab/2 = 1.1893 * 10^{-8}$ /meter, the receiver at is at altitude h_{rcvr} and emitter at h_{emit}.

Table 5. Absolute Values of Estimates of Range Error from Link-16 Operational Compensation Program and Atmospheric Filter (after 20 sec of processing of the Kalman Filter)

Estimates of Range Error for Range Values (S) from Link-16 and Atmospheric Filter (meters)																
h_{rcvr} (km)	h_{emit} (km)	S = 150 km (81 nm)			S = 250 km (135 nm)			S = 350 km (189 nm)			S = 450 km (243 nm)			S = 550 km (297 nm)		
		L16	RE	FE	L16	RE	FE	L16	RE	FE	L16	RE	FE	L16	RE	FE
0	5.2	6	8.8	0.1	8	0.7	1.0	*	*	*	*	*	*	*	*	*
	10	3	4.2	2.7	3	2.0	2.0	0	1.3	1.5	*	*	*	*	*	*
	15	1	5.3	1.0	4	3.5	0.5	9	2.9	2.6	19	2.3	8.9	*	*	*
5.2	5.2	4	5.2	1.3	5	4.0	5.0	2	2.8	3.6	5	2.9	2.1	20	3.5	3.5
	10	0	7.6	6.3	2	7.0	9.0	6	3.7	5.9	14	3.9	8.8	28	3.3	1. 6
	15	6	4.5	6.2	11	6.0	8.0	17	6.4	10.4	28	6.6	11.8	43	5.7	5.4
10	10	5	5.0	4.9	9	4.0	6.0	16	3.9	8.5	25	4.4	7.1	30	6.3	2.9
	15	11	4.3	3.3	22	3.0	8.0	29	3.4	7.2	40	3.6	6.5	57	4.2	5.1
15	10	11	2.6	5.3	22	1.0	12.0	30	4.0	11.7	39	5.6	12.3	58	5.4	12.3
	15	17	1.0	3.3	21	0.2	8.0	38	2.0	6.3	51	1.4	11.0	65	3.5	8.6
20	15	23	4.0	0.5	33	2.3	3.8	50	1.9	5.7	63	1.2	8.9	87	2.9	2.6
	20	35	4.0	2.5	48	3	2.0	65	2.7	2.3	91	2.0	2.3	109	1.2	1.3

Table Notes:
- This table displays Atmospheric Filter results for the FE and RE models, as labeled. The results labeled "L16" are from Reference 4, Table 9, page 23, and are obtained using the currently implemented Link-16 compensation model.
- An asterisk, *, indicates that the emitter is below the receiver's horizon.
- There are three L16 units emitting data in the Atmospheric Filter results. Unit 1 is the receiver at altitude h_{rcvr} and h_{emit} is that of emitter Unit 3. Units 1, 2 and 4's positions are fixed at the latitudes, longitudes & heights shown; for unit 3, the values represent its position:

L16 platform No.	Latitude (deg)	Longitude (deg)	Altitude (m)
1	45.0	45.0	h_{rcvr}
2	46.0	47.0	25000.0
3	46.85	46.85	h_{emit}
4	44.9	46.0	20000.0

- S = the true range between receiver and emitter. The latitude & longitude values for units 1 and 3 shown are for the particular case of S ≈ 250 km. Unit 2 is about 193 km from unit 1 and 4 is about 82 km from 1 for all values of S. The EKF results for 1 and 3 will vary somewhat depending upon where units 2 and 4 are relative to 1 and 3. The values of S, h_{rcvr} and h_{emit} were chosen to compare to Table 9 in Reference 4, page 23.

Here, the "home" or receiving Link-16 unit receives TOA information from just one other Link-16 platform transmitting PPLI messages to it every second. Each scenario simulates 90 seconds of processing of the Kalman filter. Two factors predominate in these results, relative angle and the true range between emitter and receiver.

Detailed results of the AF simulations are presented next.

6.3.1 Detailed Results

Characteristic AF results are shown in this section. In these, the range and elevation angle between the Link 16 receiver and one emitter are varied over a wide range to give characteristic results.

The characteristics of atmospheric refractivity are expected be roughly the same throughout a reasonably large volume of the atmosphere surrounding the receiving unit. How large a volume is to be determined and is also expected to vary during the course of a day and in different regions of geography. One would run several simultaneous instances of the AF, each determining the refractivity coefficients for a specific cluster of Link-16 emitters in a particular volume of the atmosphere which is characterized by one set of values of the refractivity parameters. One would also use whichever of the AF models (FE or RE) works best for a given geometry.

The AF must also not add too much to the processing burden of the existing OCP to determine these in real time. Therefore, the maximum number of sectors to divide the atmosphere into and still obtain reasonable results from the Filter without unduly burdening the OCP should be determined in the next phase of this study. On the other hand, which model (FE or RE) to use in a particular emitter-receiver geometry can and will be determined in this phase.

In order to determine this, the filter is run in a number of characteristic scenarios to see how the results vary. The predominate factors affecting the filter results are true range and emitter to receiver elevation angle. Examining short, intermediate and large values of range and high and low elevation angles should cover the possibilities. As the short and medium range scenarios with high elevation angle are similar, only the latter will be examined here. A very long range scenario is also included for a total of nine scenarios (see Table 7).

6.3.1.1   Introduction to the AF Test Bed

6.3.1.1.1 Overview of Tests

The objective of these tests is to determine how well the Atmospheric Filter performs in a variety of navigation scenarios and how well the RE and FE models perform relative to each other, both as to overall performance and how soon they converge to a reasonable solution.

The expectation is the FE model will perform well at shorter ranges as the earth's surface is relatively flat out to about 100 km from an observer at sea level and the FE model is simpler. Conversely, the RE model is expected to perform better at longer ranges. The elevation angle between the emitter and receiver is also of importance. Both the RE and FE models have small angle approximations and are expected to perform best at low elevation angles.

The AF program is then run for short, medium and long ranges at low and medium angles, to assess the performance. The resulting nine scenarios are described in Table 6.

As the similarity of the Exponential ("EXP") and Smith-Weintraub ("S-W") refractivity models has already been noted, it is important to see if there are any differences can be noted in the AF results using them, so the AF program is run twice for each scenario, once using each refractivity model. Nothing else is changed in a particular run when the refractivity model is changed.

6.3.1.1.2 Environment

6.3.1.1.2.1   Refractivity

The Smith-Weintraub refractivity model uses temperature, total air pressure and air pressure due to humidity as inputs. In order to use this model in the AF program, models of temperature, total air pressure and air pressure due to humidity were obtained as functions of altitude for average conditions in temperate regions. For temperature, the average behavior is to decrease linearly with altitude up to the tropopause, stay relatively constant throughout the tropopause, and then rise linearly up to the level of the stratosphere. Piecewise linear fits to this behavior as a function of altitude were used. Simple damped exponential fits were used for the pressures. These are described in Appendix A, Section 12.1, below. Thus, we were able to characterize the Smith-Weintraub refractivity as a function of altitude and so use it in the Atmospheric Filter runs.

An exponential fit to the Smith-Weintraub refractivity as a function of altitude was then obtained, yielding a surface refractivity, a, of 326×10^{-6} and altitude coefficient, b, of 1.306×10^{-4}. These were then used as the truth values of a and b for the Exponential refractivity model in the program, to perform the refractivity effects on the propagation of the Link-16 signal. The Exponential refractivity model varies with height above sea level, it needs no other input beyond the values of the parameters a and b.

The refractivity is governed by the same values of a and b everywhere in all the simulations performed in this study. Basically, the same temperate "weather" obtains throughout the volume of atmosphere in which all the Link-16 units are located. The Thayer model was not implemented as its results were shown in section 6.1.4 to be almost exactly the same as that of the Smith-Weintraub model.

6.3.1.1.2.2 Atmospheric Structure

The Link 16 PPLI signals are refracted per the RE model (q.v., Section 6.2.1.2), yielding the basic measurement for both FE and RE models: the signal's Time Of Arrival (TOA). The reason this is used is that the RE model is more accurate, yielding more realistic results.

The signals are refracted along their path of propagation in one layer of the round earth atmosphere after another, in a sequence of altitudes, $\{h_i\}$, where i = the layer number. Each layer is characterized by a constant index of refraction, $n_i = n(h_i)$, as given by the Exponential or Smith-Weintraub model at each h_i . This situation is illustrated in Figure 13. Coleman, in Reference 4, page 13, indicates that a layer thickness of dh (= $h_{i+1} - h_i$) of about 10 m works well.

6.3.1.1.3 Participants

The emitter and receiver in each simulation are assumed to be Link-16 units. The emitter sends a PPLI message to the receiver. Each simulation involves three emitters, each sending one PPLI per second to the receiver in a round robin fashion. Thus, a signal from a given emitter will be received every three seconds.

6.3.1.1.4 Atmospheric Filter: Extended Kalman Filter

The Kalman Filter parameters are:

1. Initial state vector (x) and covariance matrix (P) values
2. Noise parameters, including measurement noise matrix (R), process noise matrix (Q) values and
3. Observation Transformation Matrix (H).

6.3.1.1.5 Test Case Scenarios

In each case, the AF routine was run with three emitters near each other sending PPLIs in a round robin fashion to the receiver.

The range and elevation angles between the emitter and receiver in the scenarios are given in Table 6. Nine scenarios are considered, covering the extremes of relative positions of emitter and receiver, short range, medium range and long range, with low and high elevation angles. Note that at long enough ranges, the maximum elevation angle will be not more than about ten degrees; otherwise the emitter will be either below the receiver's horizon or above the troposphere. At any rate, in the Link-16 situation, the low angle signal paths are the norm and the higher angle results are shown to illustrate the limitations of the FE and RE models. The FE depends on the horizontal path length, l, and the RE the interior angle, ϕ. These lead to a singularity in the vertical direction with best results at low elevation angles. They could be reformulated using altitude, h, as the independent variable, but this would lead to a singularity in the horizontal direction, which is not a desirable alternative to the small angle model as the Link-16 units will tend to be at lower elevation angles to each other.

The Atmospheric Filter routine was implemented in the form of a Compaq Visual Fortran program running on a Windows 2000 PC. The source code for this program is given in Appendix D.

In each scenario, there are three emitters, each sending their PPLI messages to the receiver in a round robin fashion, once per second. Thus the data for a particular emitter is received once every three seconds. The emitters are taken to be in one squadron, thus they are placed fairly close to each other in each scenario. The results are shown for one of these Link 16 units per scenario. Neither the emitters nor the receiver move in the simulation.

In the Fortran program for each scenario, both the Round Earth (RE) and Flat Earth (FE) models are run. They are both initialized and the positions of the Link 16 units are specified at the start of the program. Next, the truth model is invoked to create the TOA (time of arrival of the Link 16 PPLI message) which is the pseudo-measurement for both RE and FE. This is obtained by refracting the signal per the Re model of atmospheric refractivity, since it is more realistic. In it, the space between emitter and receiver is broken into layers of about 10 meters height in each of which the refractivity is taken to be constant and Snell's law is used to refract the signal. The true range is also calculated. At the end f the program the range derived from the Kalman states of the refractivity are then used to create the AF estimate of the range. This is differenced from the "true" range and a plot of this estimate as a function of processing time is shown for each scenario.

The filter is run for 90 seconds of simulation time. There are three Link 16 units in each test; they send PPLIs to the receiver in a round robin fashion such that the receiver receives PPLIs from a given emitter once every three seconds. Thus, there are thirty PPLIs received from a given emitter in each simulation run.

The other two plots presented per run show the EXP states a and b from the FE and RE models. These plots show how observable the states are, if the state does not change value with processing time, it is probably unobservable for that geometry.

In each case, the initial Kalman model values selected are those which are seen to give the best performance for each model These values are the initial state covariances (P matrix), the measurement error noise covariances (R matrix) and state process noise covariances (Q matrix). The only measurement noise present is the quantization of the TOA in units of 12.5 nanoseconds. How best to vary the initial conditions is a part of the present stage of development of the AF.

Table 6 Scenario Parameters

Test Case Scenario			Range (km)	Emitter-Receiver Elevation Angle (degrees)	Emitter Altitude (m)	Receiver Altitude (m)
Number	Range	Elevation (emitter relative to receiver)				
1a	Short	Low (both at medium altitude)	9	3	10500	10000
1b	Short	Low (both at low altitude)	7	3	500	100
2	Short	Medium	9	10	10000	10000
3	Short	High	5	45	15000	10000
4	Medium	Low	60	1	26000	24500
5	Medium	Medium	60	-10	25000	23600
6	Medium	High	22	45	26000	0
7	Long	Low	145	0.5	12000	9000
8	Long	"High"	225	10	45100	10000
9	Very Long	Low	475	3	45000	25000

6.3.1.2   Test Case Number 1: Short range, low elevation angle

6.3.1.2.1 Overview

Specific Objectives

Determine if there is any difference if the emitter and receiver are low in the atmosphere where the index of refraction is generally greater than higher in the atmosphere. Basically, the signal should be delayed more in the lower atmosphere. So, for this test case, two runs will be examined, one at medium altitude and one at low altitude, at the same range and same relative elevation.

Specific Conditions

Emitter- receiver: range = 9 km, elevation = 3 degrees. Refer to Table 6: "Scenario Parameters" for other details.

Performance Expectations

Both FE and RE should work well here because of the low emitter/receiver relative elevation angle.

The Exponential (EXP) and Smith-Weintraub (S-W) results are expected to be very similar in this and most of the other scenarios, so only any differences will be noted. The Smith-Weintraub refractivity used here is derived from average conditions. It is expected that more realistic Smith-Weintraub refractivities derived from actual meteorological data from specific locations in the world will give somewhat different results at low altitude.

The FE and RE results are expected to be somewhat better at higher altitudes as there is less refractivity to correct there.

6.3.1.2.2 Medium Altitude Case (1a)

6.3.1.2.2.1 Exponential Model Results
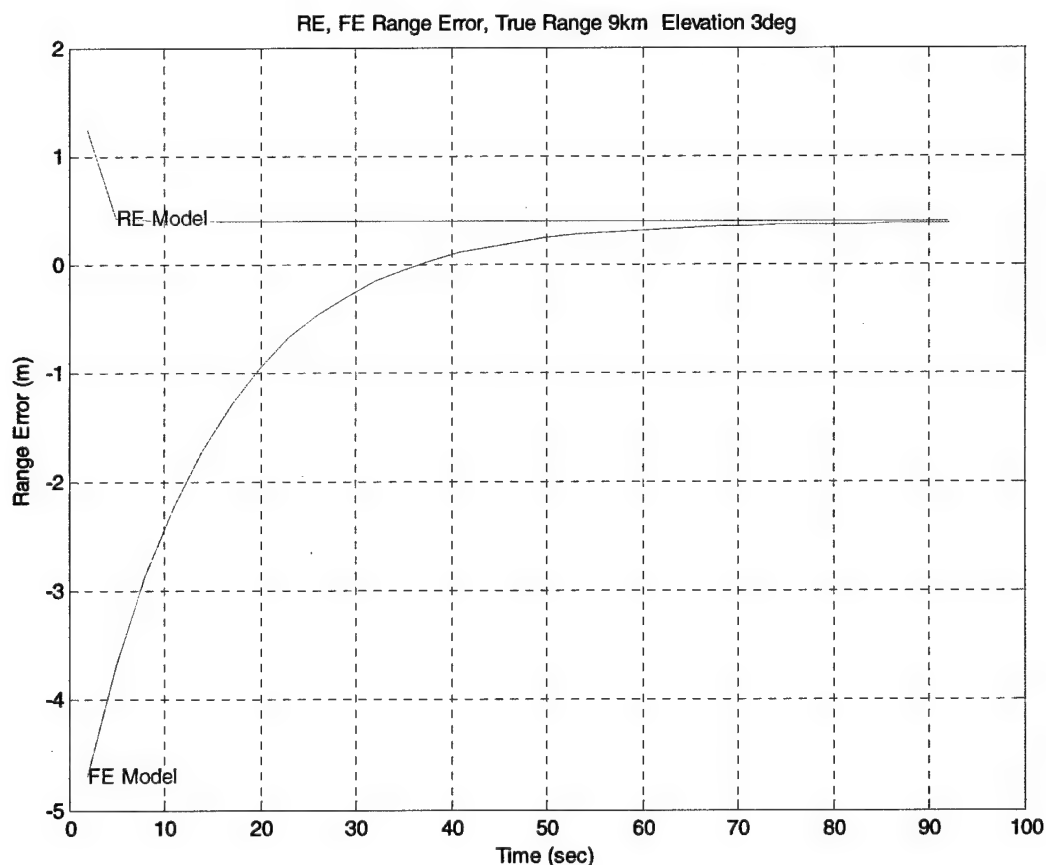
Figure 22. EXP Range Error, Short Range, Low Angle Medium Altitude
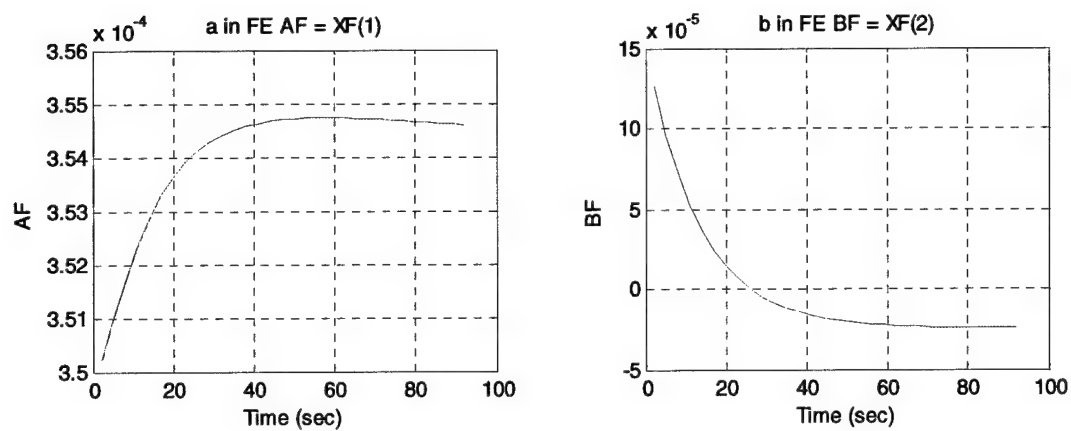
Figure 23 EXP FE refractivity coefficients Short range, low angle Medium Altitude
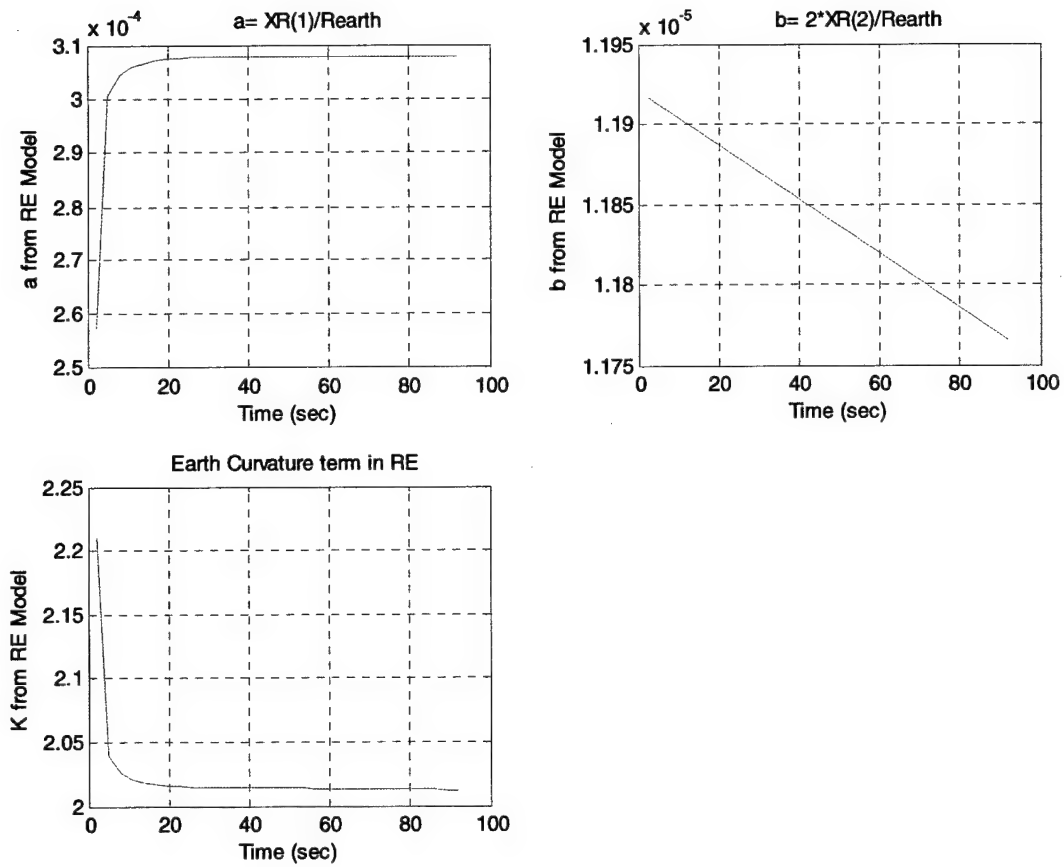
Figure 24 EXP RE refractivity coefficients Short range, low angle Medium Altitude

6.3.1.2.2.2   Smith-Weintraub Model Results

RE, FE Range Error, True Range 9km   Elevation 3deg
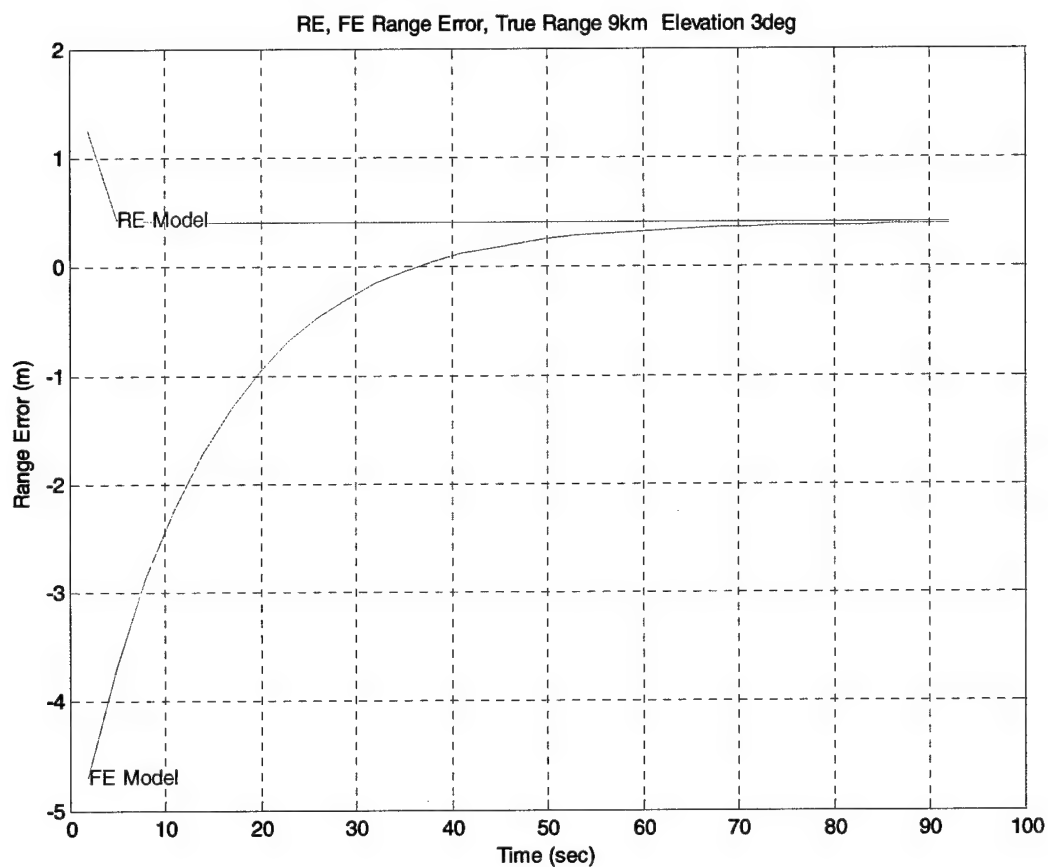
Figure 25 SW Range Error: short range, low angle at Medium Altitude
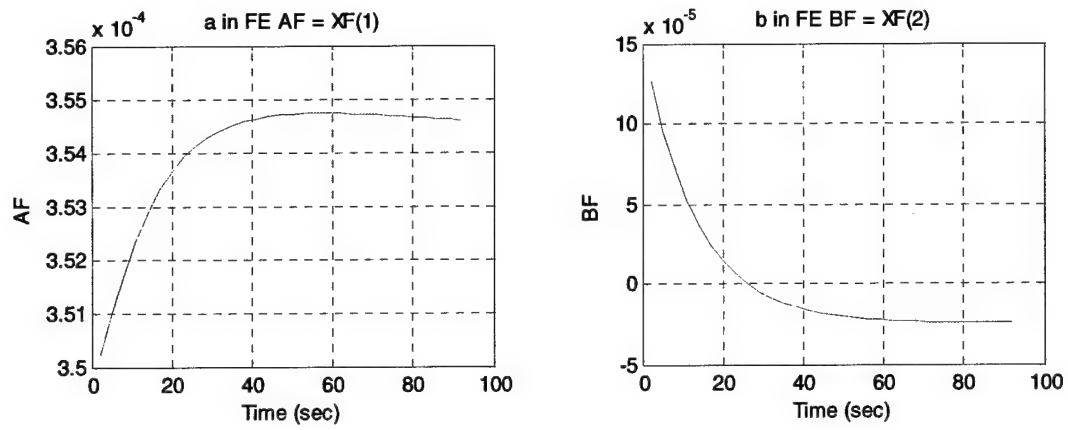
Figure 26 SW FE refractivity coefficients short range, low angle at medium Altitude
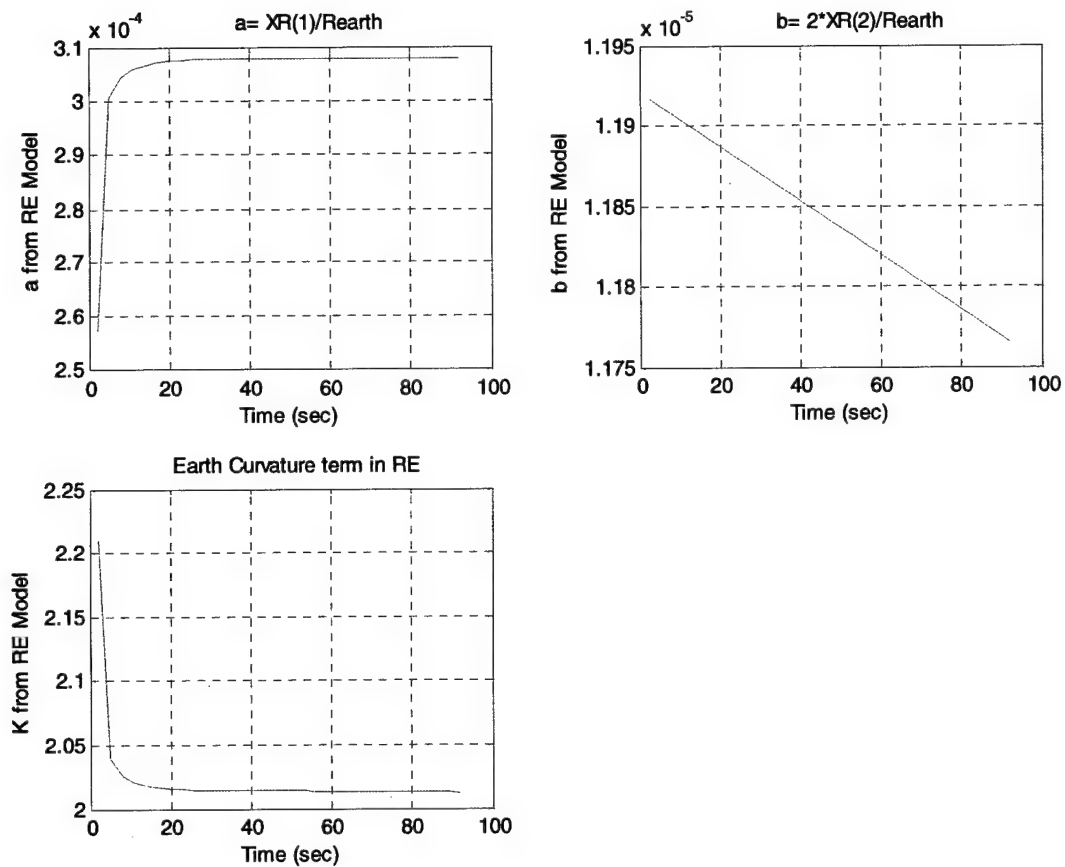
Figure 27 SW RE refractivity coefficients short range, low angle at medium altitude

6.3.1.2.3 Low Altitude Case (1b)

6.3.1.2.3.1   Exponential Model Results

RE, FE Range Error, True Range 9km Elevation 3deg
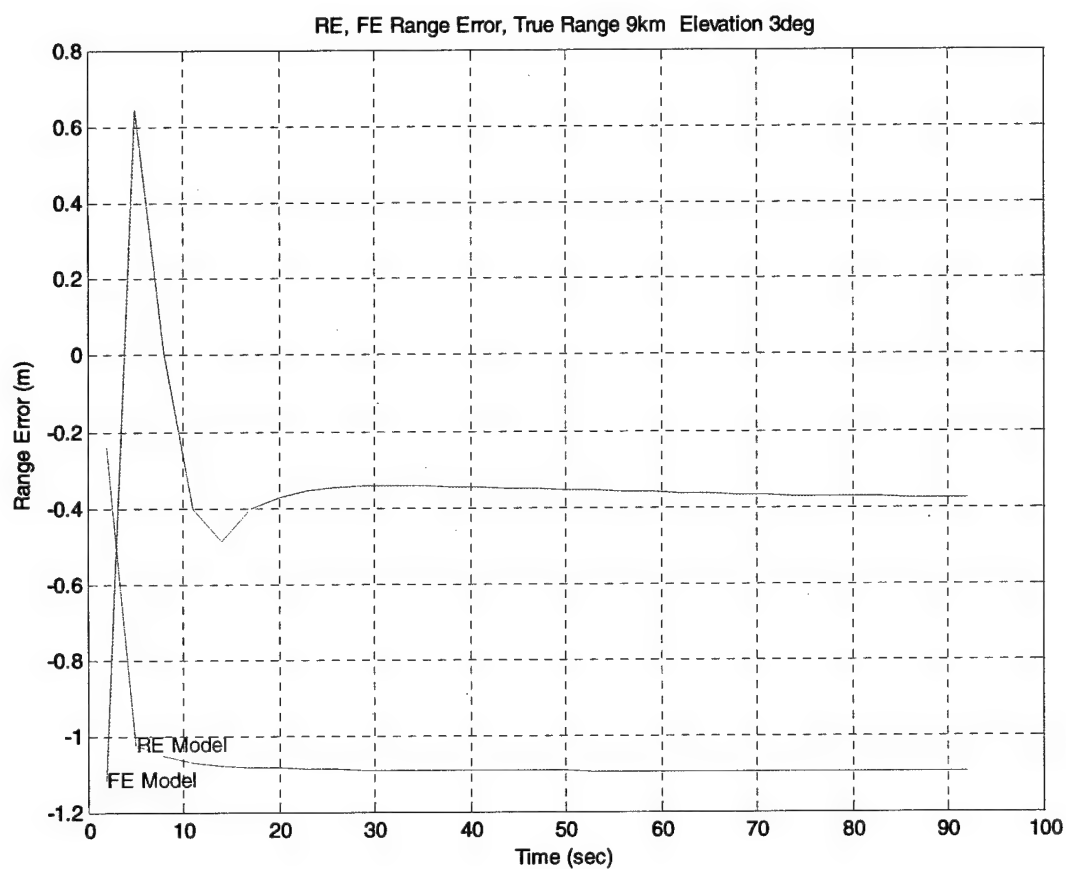
Figure 28 EXP Range Error Short Range, Low Angle at Low Altitude
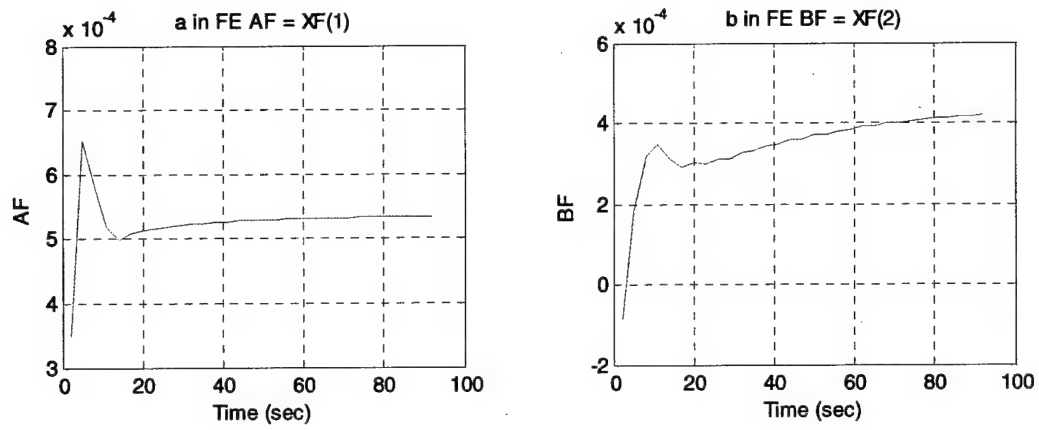
Figure 29 EXP FE refractivity coeffs Short Range, Low Angle at Low Altitude
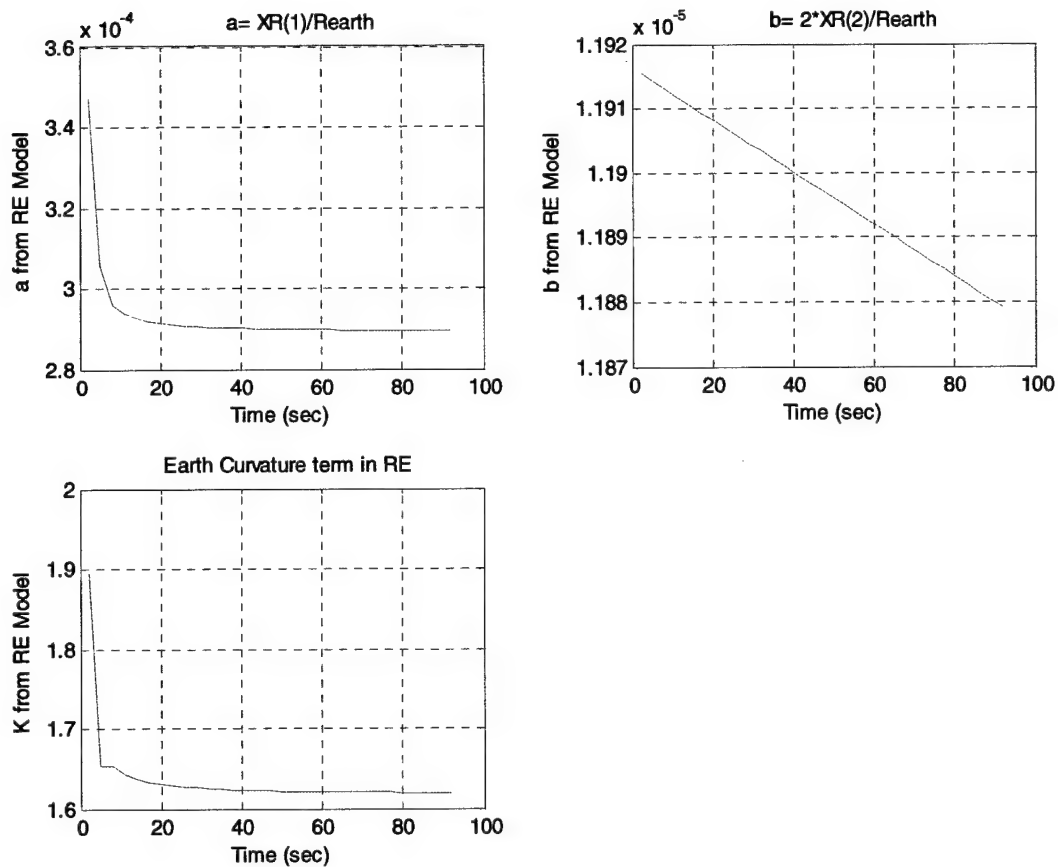
Figure 30 EXP RE refractivity coefficients Short Range, Low Angle at Low Altitude

6.3.1.2.4 Smith-Weintraub Model Results, Low Altitude

RE, FE Range Error, True Range 9km Elevation 3deg
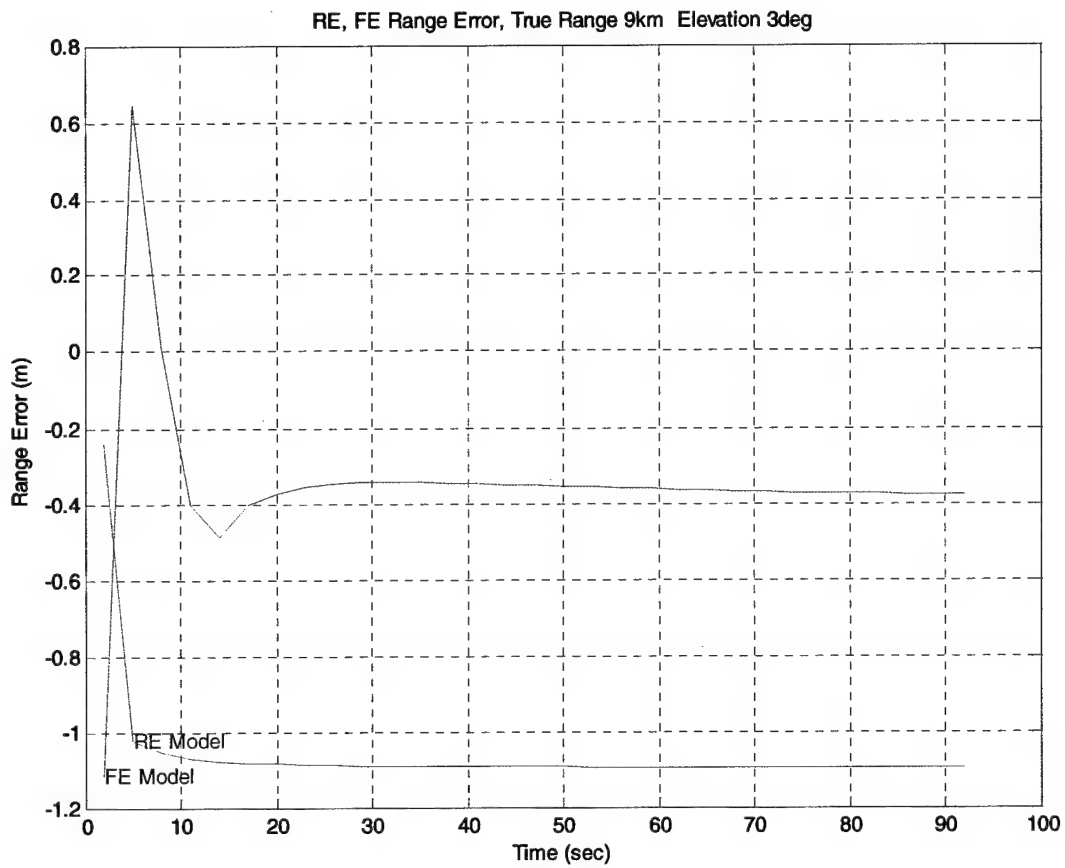
Figure 31 SW Range Error: short range, low angle at low altitude
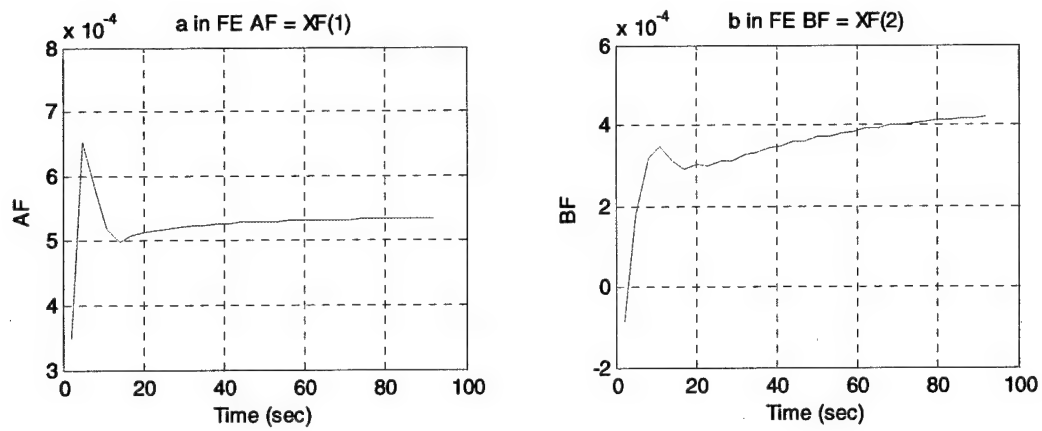
Figure 32 SW FE refractivity coefficients short range, low angle at low altitude
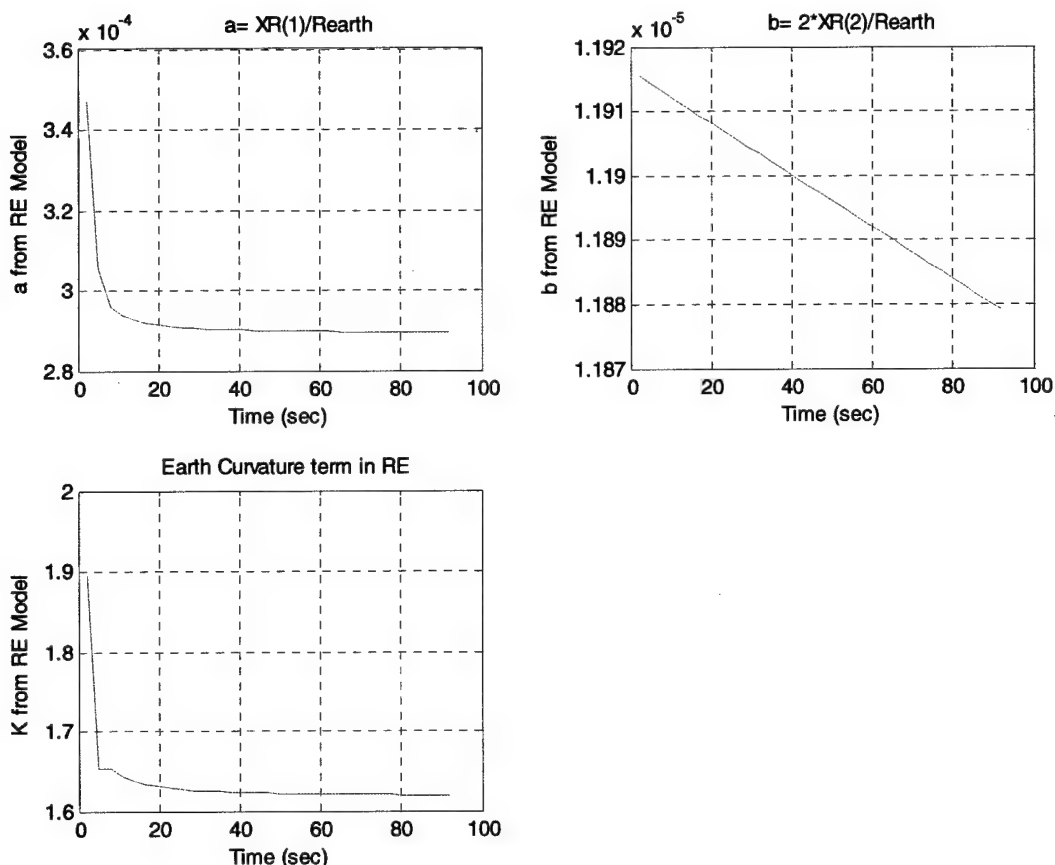
Figure 33 SW RE refractivity coefficients; short range, low angle at low altitude

6.3.1.2.5 Discussion

As expected, the Exponential and Smith-Weintraub results are very similar. The range errors are low in both the low and medium altitude cases.

The RE range error converges more rapidly than that of the FE at medium altitude, both converge rapidly at the lower altitude, where the refraction is stronger.

In the medium altitude case, the FE converges gradually in about 50 seconds to a range error of about 0.4m. In the same time, it converges to a value of the refractivity coefficient, a, of about $3.5 * 10^{-4}$ and b of about $-2.0 * 10^{-5}$ per meter. The RE range error converges rapidly to about the same value, 0.4 meters, and coefficient a to about $3.1 * 10^{-4}$, with curvature factor k to about 2.0. However, coefficient b does not converge, but varies linearly with time through a small range around a value $1.1 * 10^{-5}$ per meter.

In the low altitude case, the FE gives a smaller range error than the RE but RE converges more rapidly. The FE range error overshoots zero from about -1 m and then converges to -0.4 m range error. Its value of coefficient a similarly overshoots to settle at about $5 * 10^{-4}$ and b to $4 * 10^{-4}$ per meter. The RE converges rapidly to an error of about -1.1 meter. The coefficient, a

converges from above to about $2.9 * 10^{-4}$ and k to about 1.6. Coefficient b, is small and behaves as in the medium altitude case, perhaps reflective of the fact that in the short range, only a fairly small slice of the atmosphere is sampled.

6.3.1.3   Short Range, Medium Angle Case

6.3.1.3.1 Overview

Specific Objectives

To see how the models perform at a higher elevation angle.

Specific Conditions

Emitter- receiver: range = 9 km, elevation = 10 degrees, altitudes about 25 km.   Refer to Table 6: "Scenario Parameters" for other details.

Performance Expectations

Performance for both models should be not as good as for a lower angle. The EXP and S-W results should be similar.

6.3.1.3.1.1 Exponential Model Results

RE, FE Range Error, True Range 9km Elevation 10deg
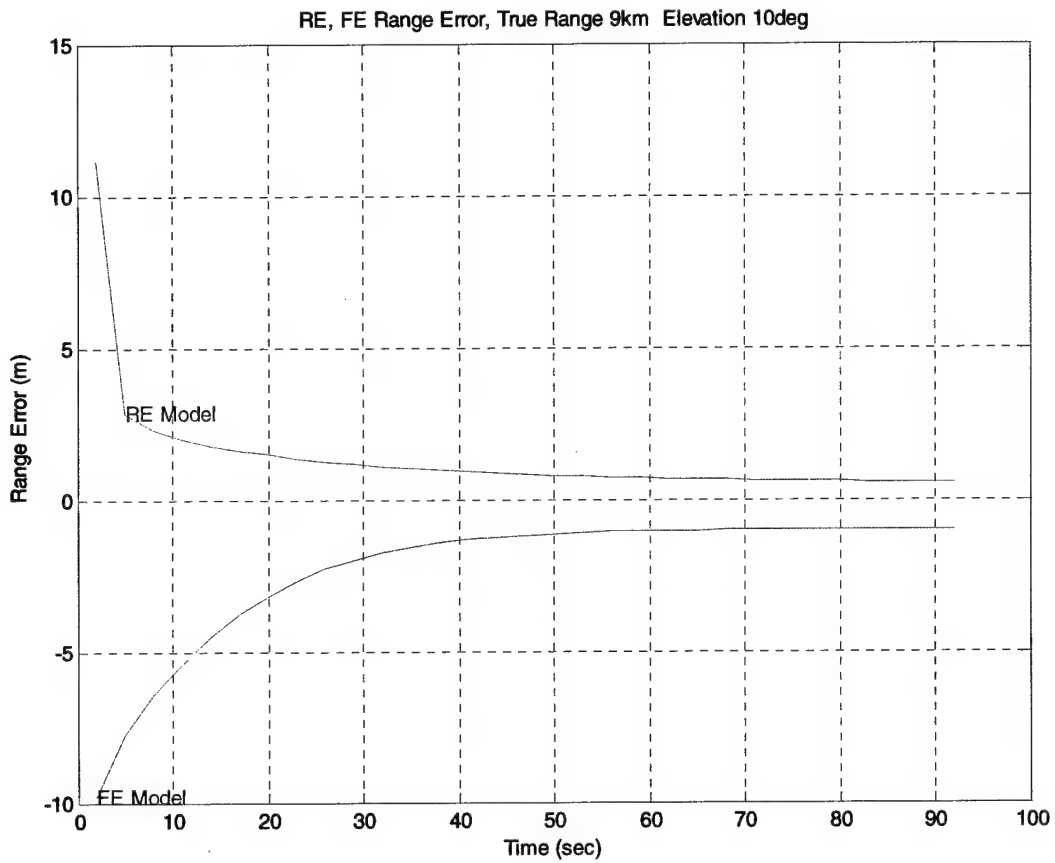
Figure 34 EXP Range Error; Short Range, Medium Angle
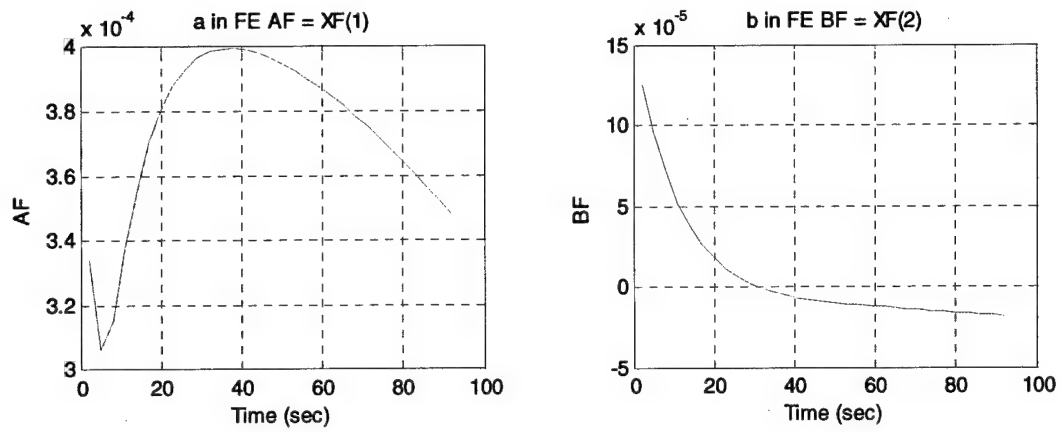
Figure 35 EXP FE refractivity coefficients Short Range, Medium Angle
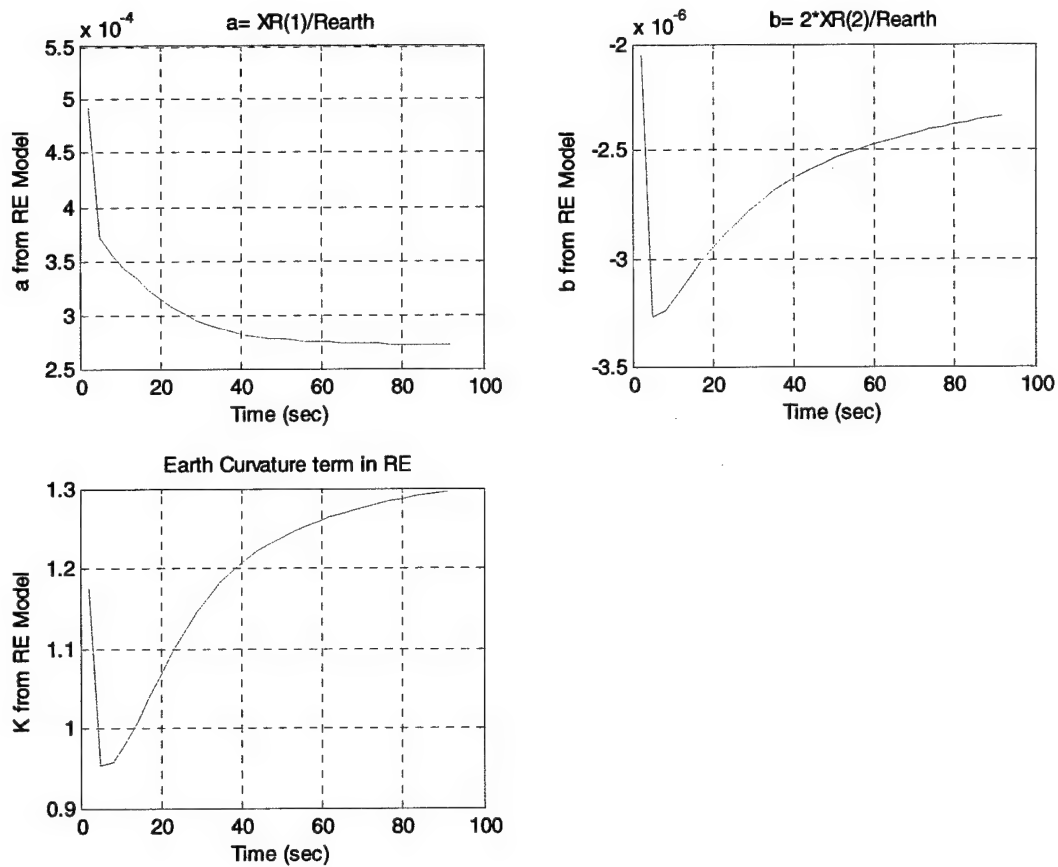
Figure 36 EXP RE refractivity coefficients Short Range, Medium Angle

6.3.1.3.1.2 Smith-Weintraub Model Results

RE, FE Range Error, True Range 9km Elevation 10deg
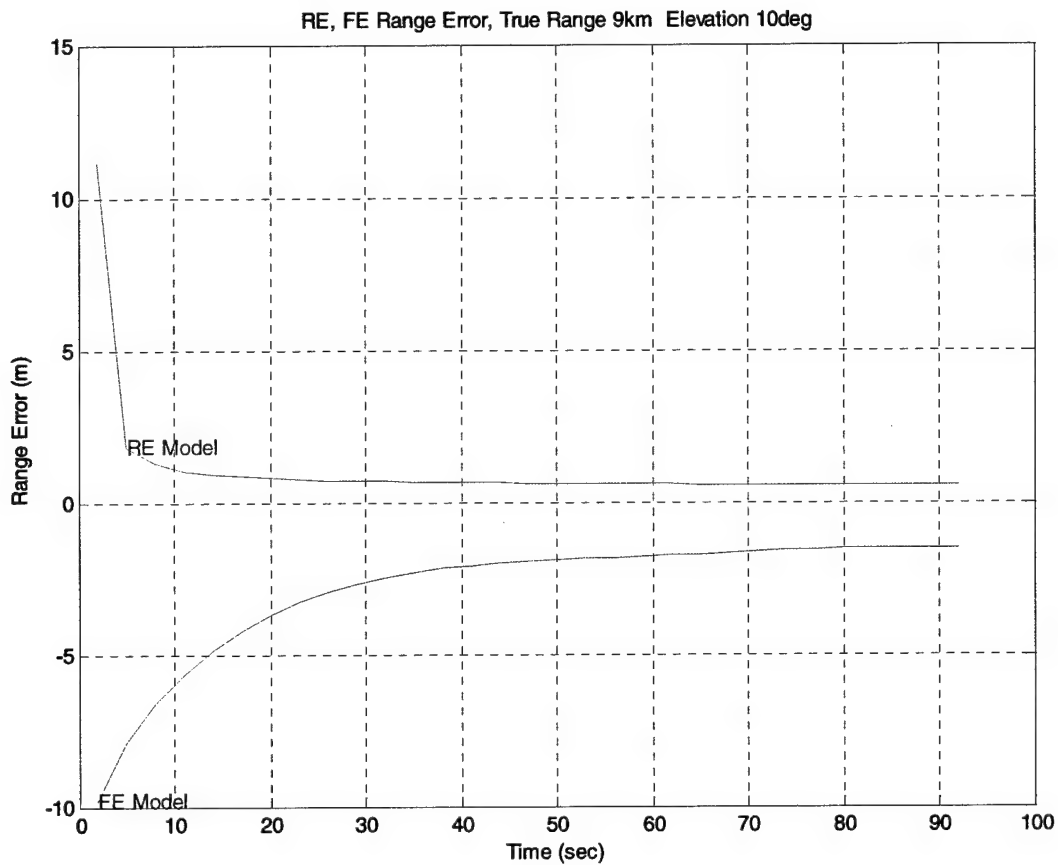
Figure 37 SW ERRS Short Range, Medium Angle
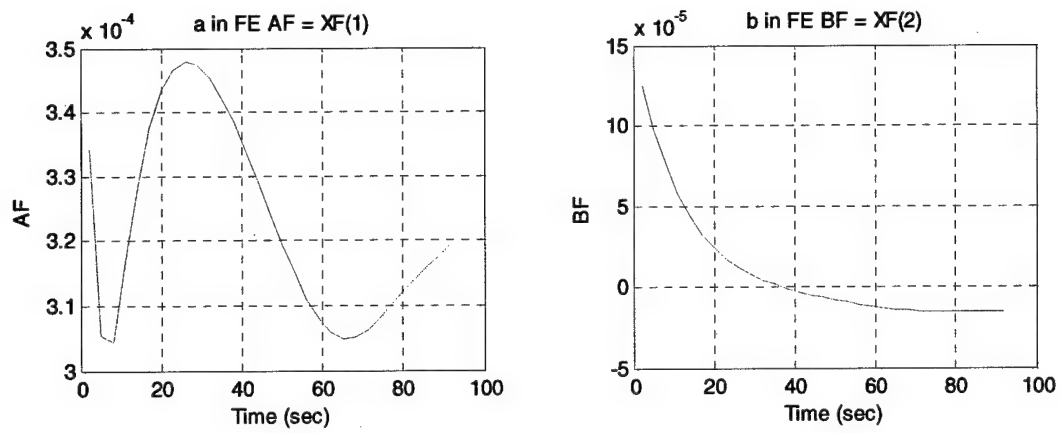
Figure 38 SW FE refractivity coefficients Short Range, Medium Angle
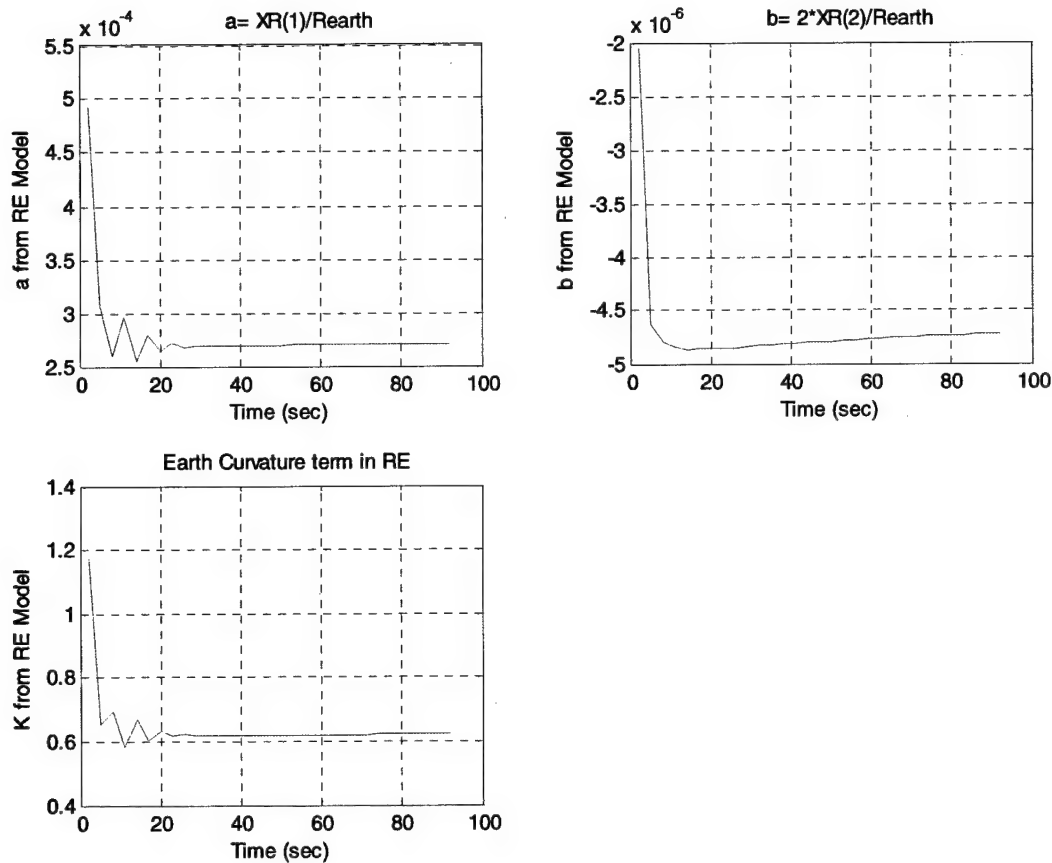
Figure 39 SW RE refractivity coefficients Short Range, Medium Angle

6.3.1.3.2 Discussion

Surprisingly, the EXP and S-W models give somewhat different results here.

Using the EXP model, the RE range error converges more rapidly than that of the FE, but to a slightly higher value of somewhat less than 1, from above. From below, the FE ranger error converges in about the same way as in the low angle, short range, medium altitude case, but to a negative range error of about -1 meter. In the FE, the coefficient a rises to about $4.0 * 10^{-4}$ but then drops through $3.4 * 10^{-4}$ which not apparently converging. Its b value converges to about the same value as in the previous test case, however. In the RE, the refractivity coefficient a converges to about $2.7 * 10^{-4}$, curvature factor k to about 1.3 while b is negative and small, about $-2.3 * 10^{-6}$, indicating a somewhat linear refractivity behavior with altitude.

In the S-W model results, while the RE and FE range errors are about the same and in the EXP case, the refractivity coefficients behave differently. In the FE, coefficient b is similar to that of the previous case, but a oscillates in time about the value $3.2 * 10^{-4}$. In the RE, coefficients a and b and curvature factor k all converge rapidly to values of $2.6 * 10^{-4}$, $-4.8 * 10^{-6}$ /m and 0.6, respectively.

6.3.1.4   Short Range, High Angle Case

6.3.1.4.1 Overview

Specific Objectives

To see how the performance changes for a higher angle.

Specific Conditions

Emitter- receiver: range = 5 km, elevation 45 degrees, altitude about 10 km. Refer to Table 6: "Scenario Parameters" for other details.

Performance Expectations

Performance for both models should be not as good as for a lower angle, in fact, they should not be very good at all. The EXP and S-W results should be similar.

6.3.1.4.1.1   Exponential Model Results

RE, FE Range Error, True Range 5km   Elevation 45deg
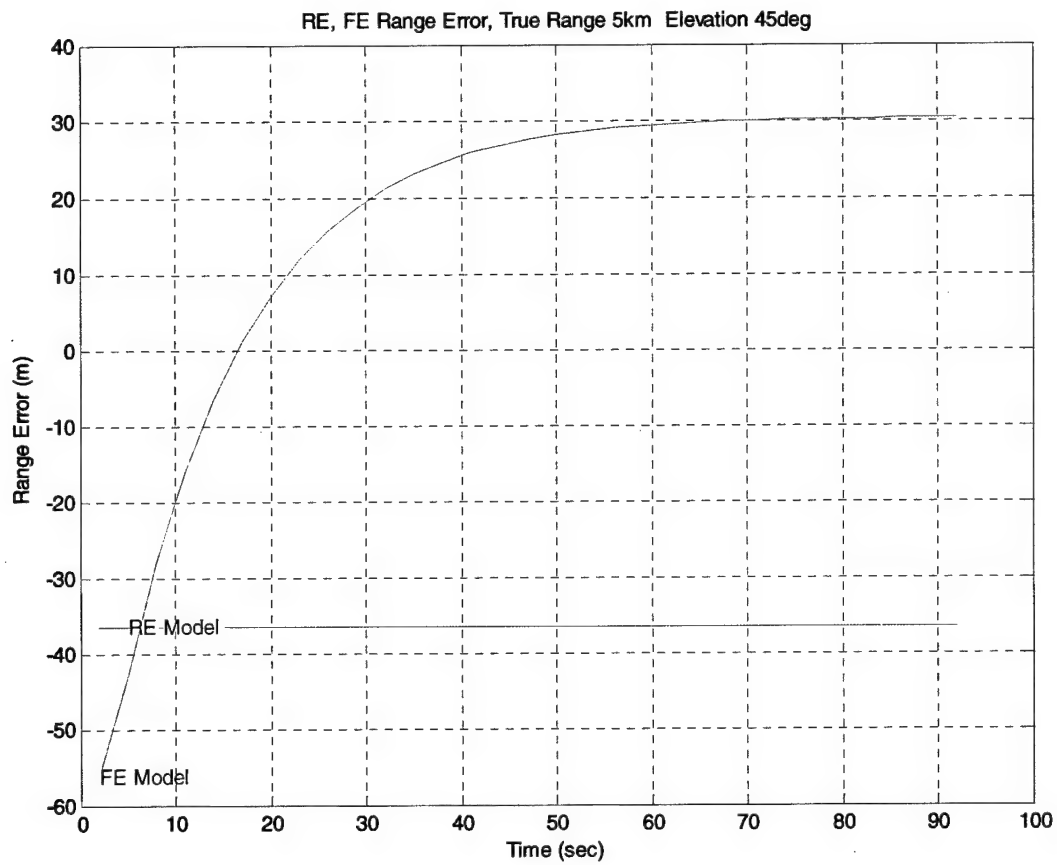
Figure 40 Exp Range Error Short range High Angle
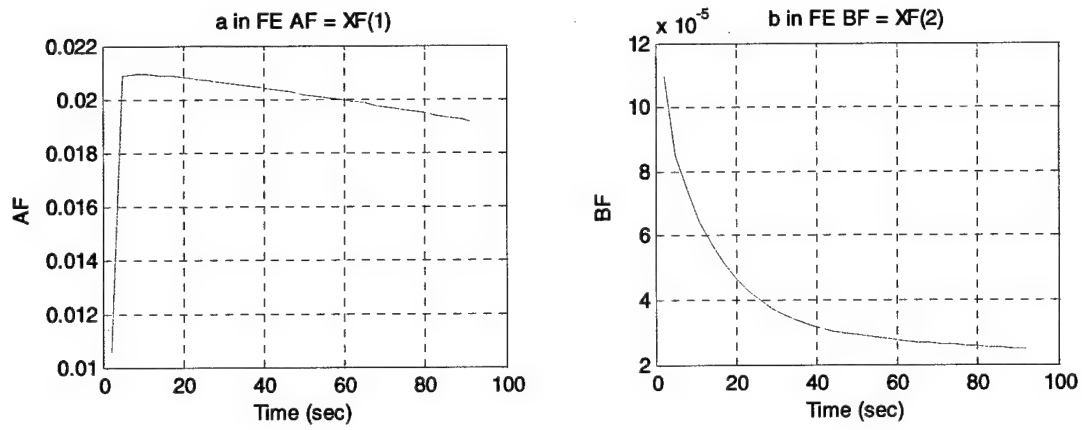
Figure 41 Exp FE refractivity coefficients Short range High Angle
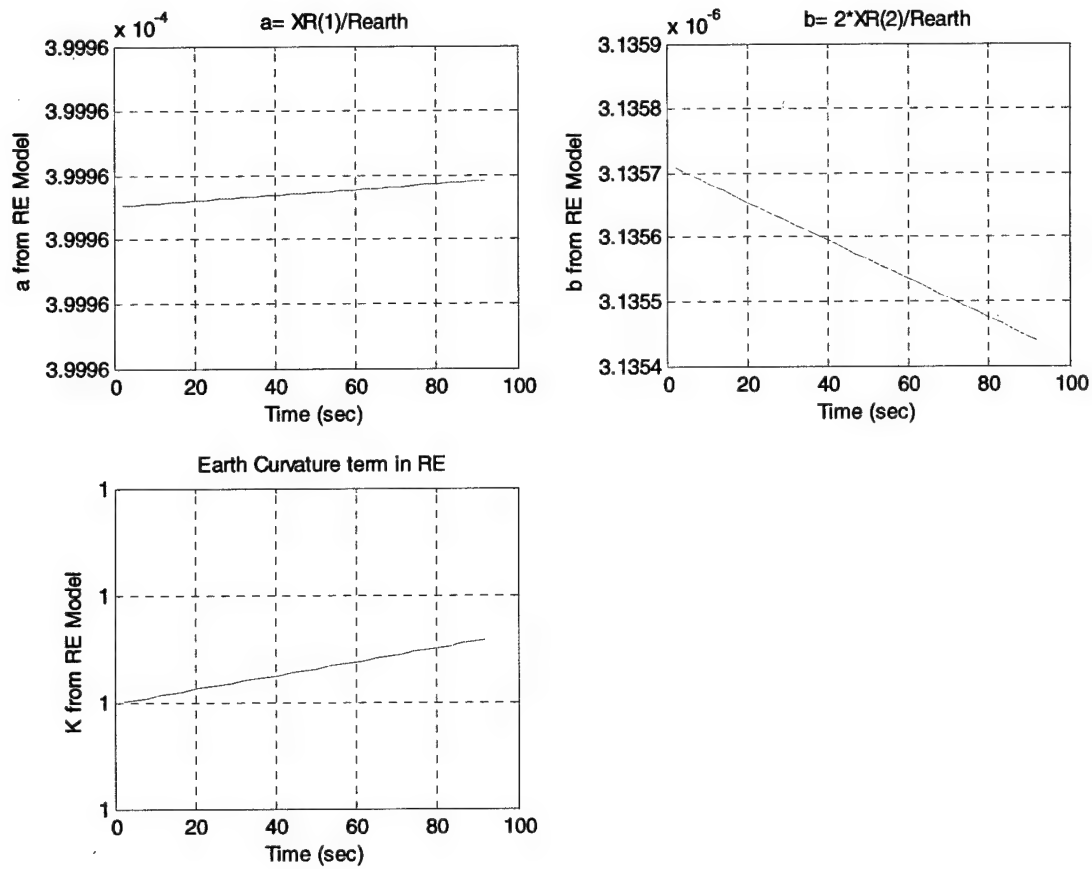
Figure 42 Exp RE refractivity coefficients Short range High Angle

6.3.1.4.2 S-W Model Results

RE, FE Range Error, True Range 5km   Elevation 45deg
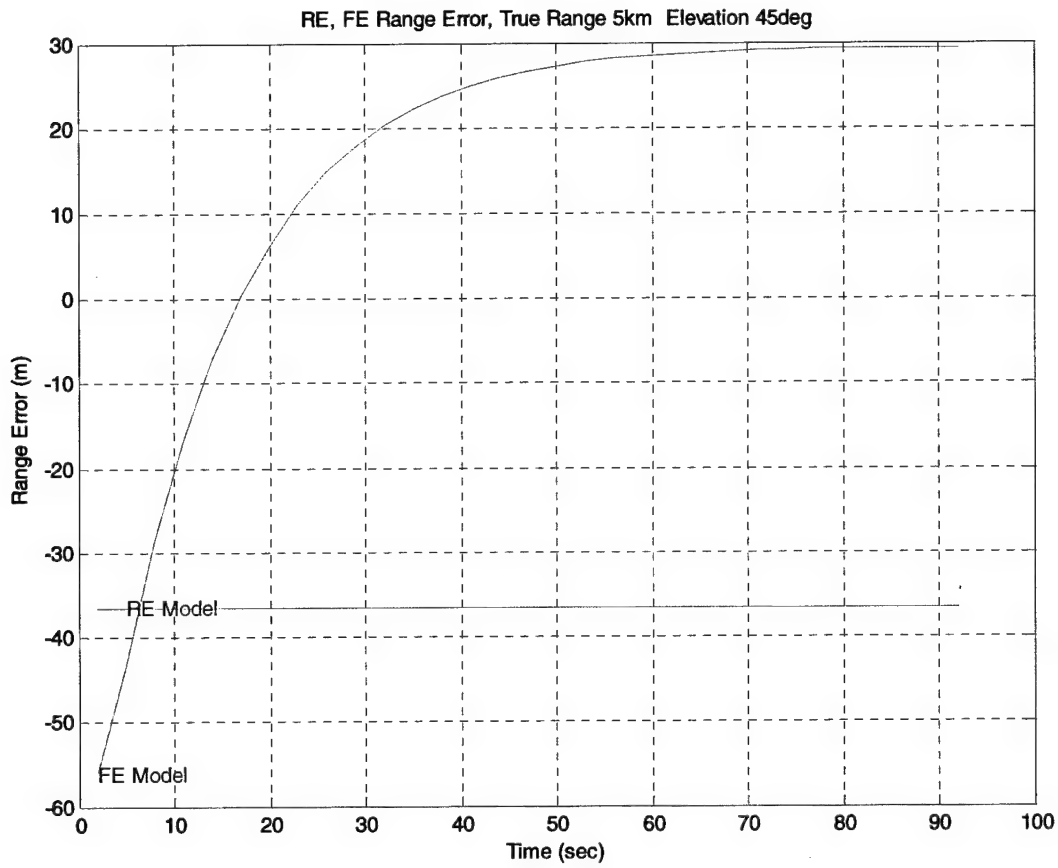
Figure 43 SW Range Error Short range High Angle
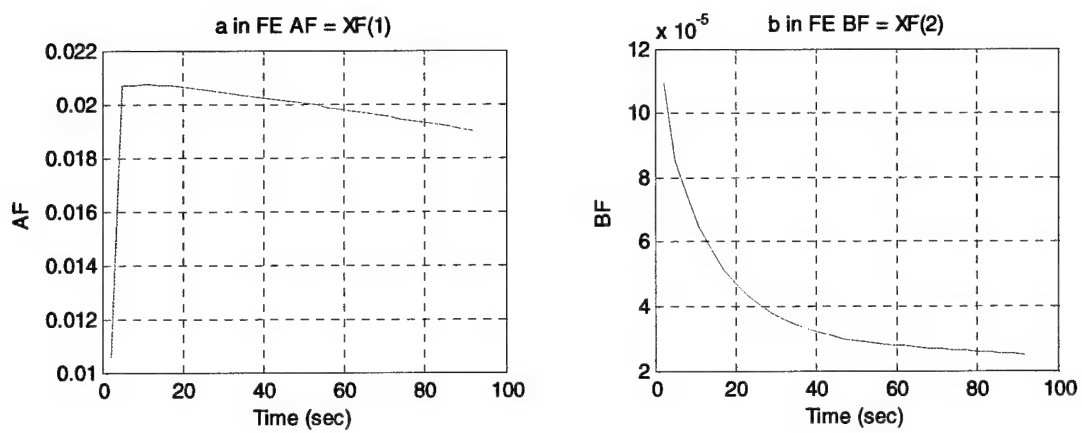
Figure 44 SW FE refractivity coefficients Short range High Angle
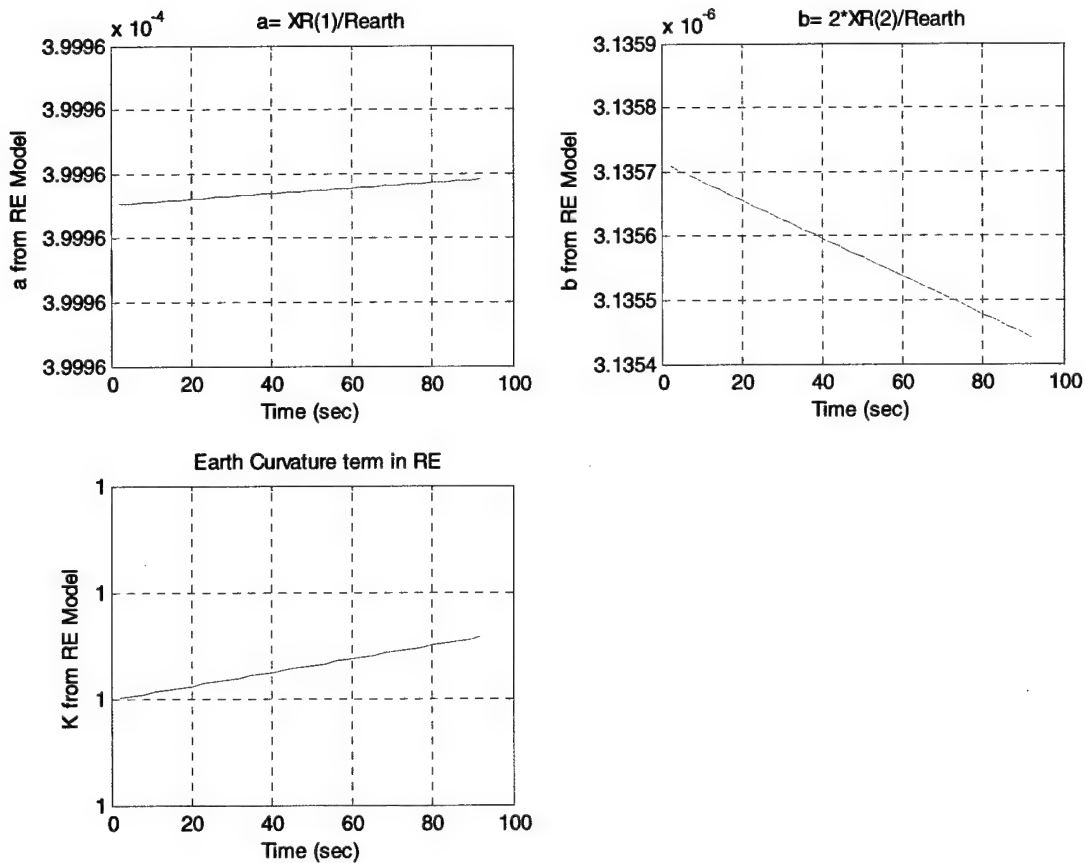
Figure 45 SW RE refractivity coefficients Short range High Angle

6.3.1.4.3 Discussion

As expected, neither model does well here and the Kalman states are not well observable. The FE range errors converge slowly to 30 m and those of the RE are stuck at about -38 m. In the FE, coefficient a is too large and b too small. The RE a and b values do not converge at all, neither does k.

The performance would be improved with a model which does not have a small angle approximation, although this may be intractable analytically. However, Link-16 units generally would operate under smaller elevation angles.

6.3.1.5   Medium Range, Low Angle Case

6.3.1.5.1 Overview

Specific Objectives

To see how the models perform at a longer range.

Specific Conditions

Emitter- receiver: range = 60 km, elevation 1.0 degrees, altitude 25 km. Refer to Table 6: "Scenario Parameters" for other details.

Performance Expectations

The EXP and S-W results should be similar. Both the RE and FE should do well, as the elevation angle is low, with the RE converging faster, as in previous test cases.

6.3.1.5.1.1 Exponential Model Results
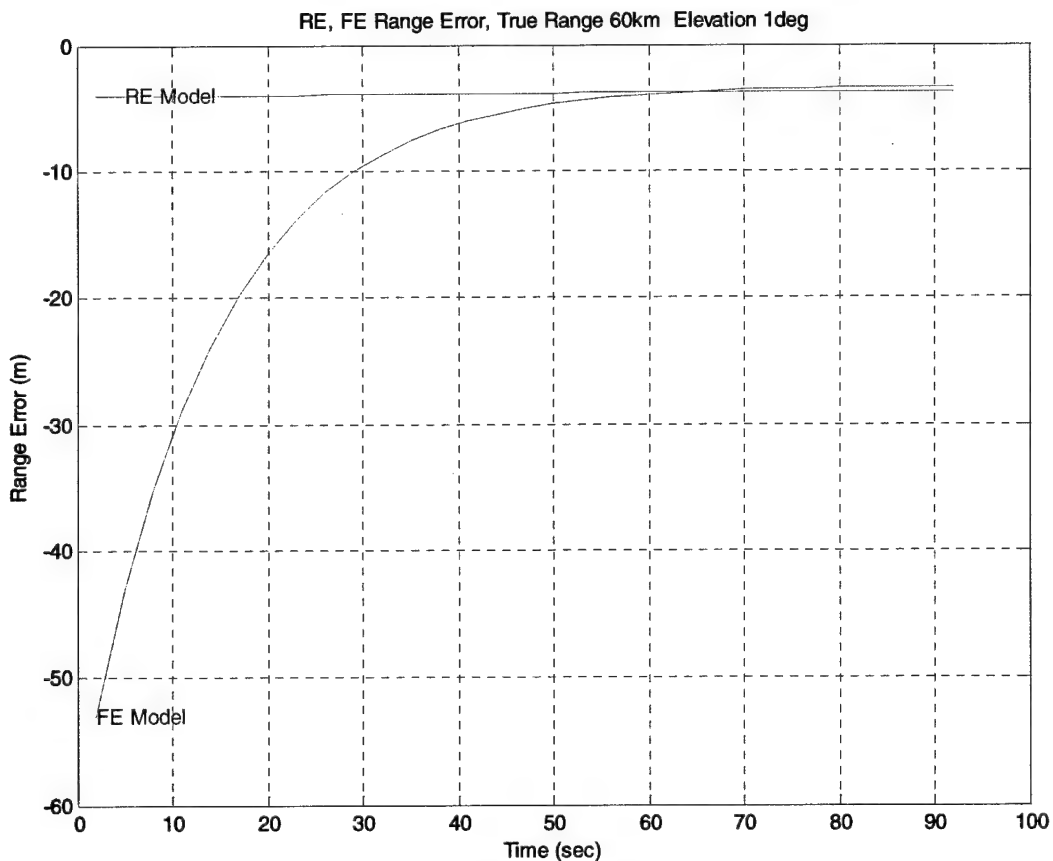
Figure 46 EXP Range Error Medium Range, Low Angle
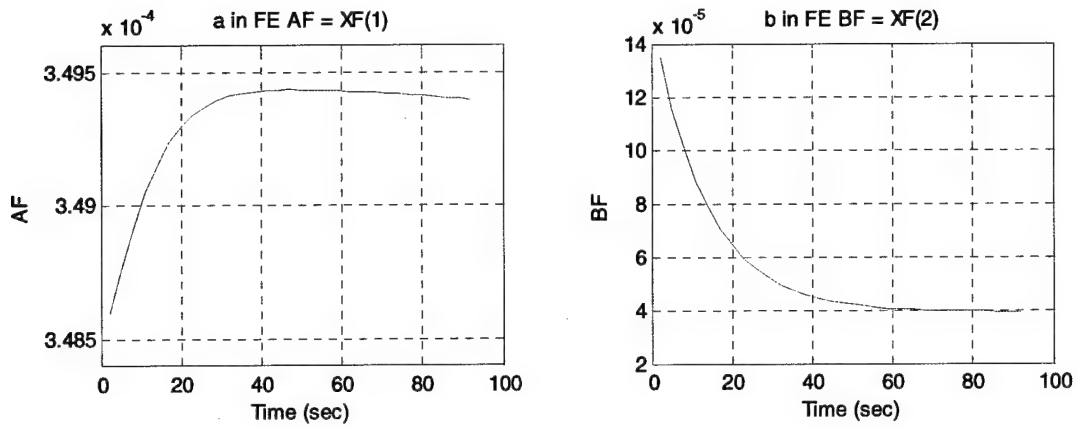
Figure 47 EXP FE refractivity coefficients Medium Range, Low Angle
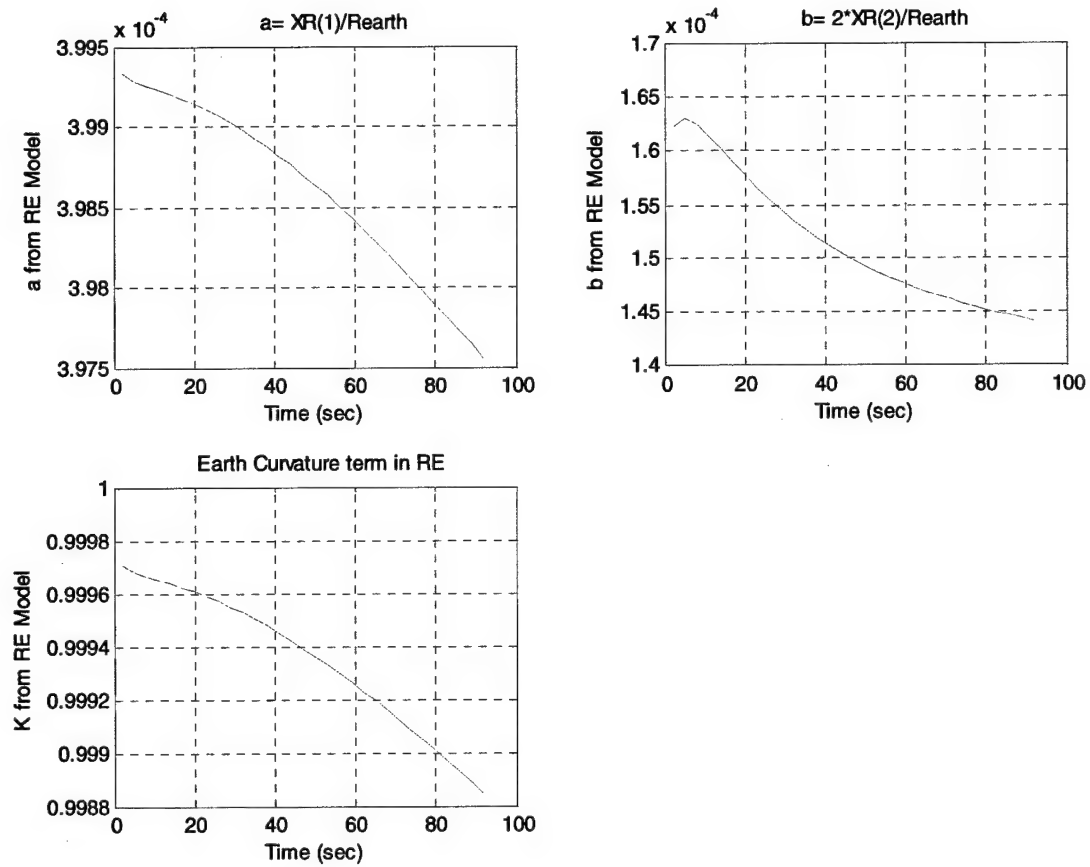
Figure 48 EXP RE refractivity coefficients Medium Range, Low Angle

6.3.1.5.1.2 Smith-Weintraub Model Results

RE, FE Range Error, True Range 60km Elevation 1deg
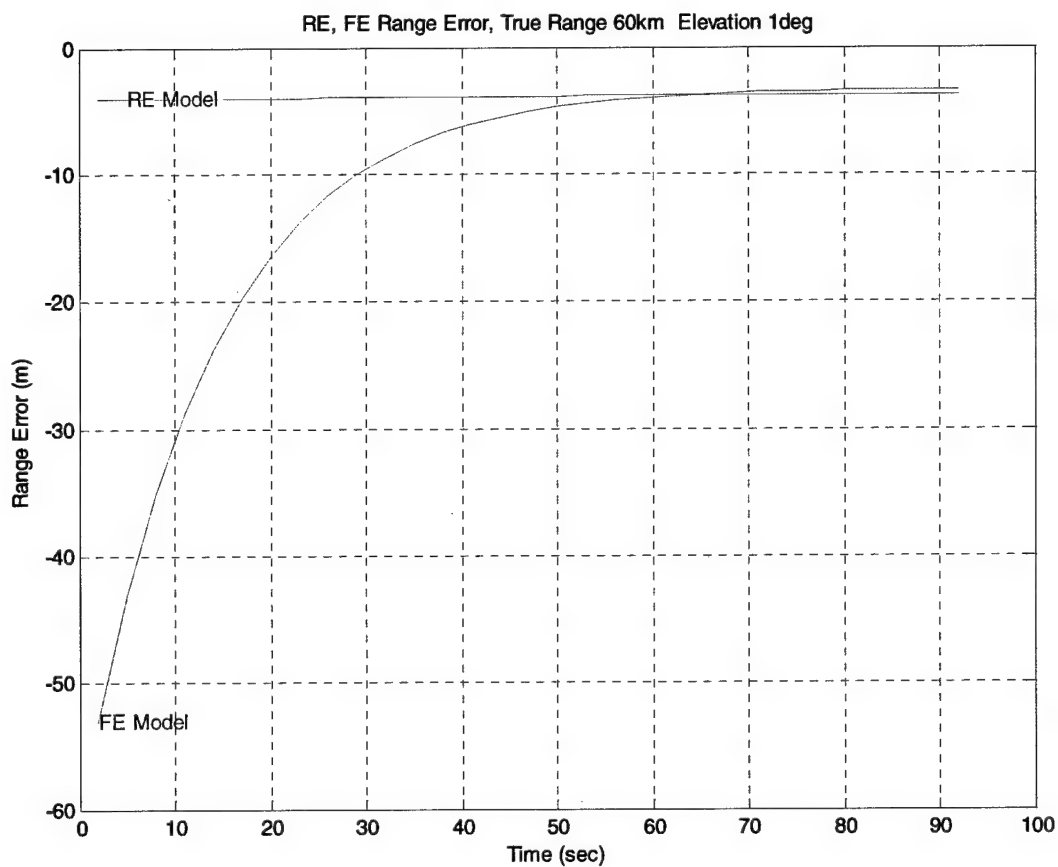
Figure 49 SW Range Error Medium Range, Low Angle
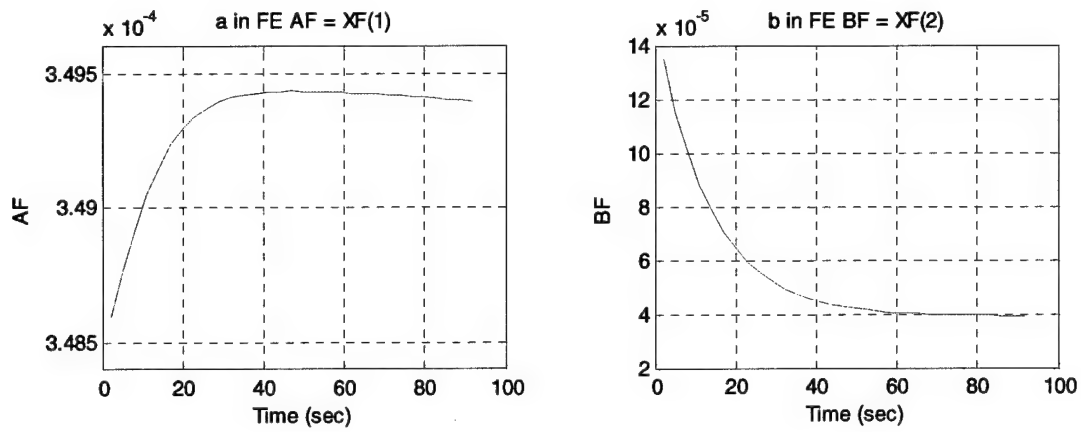
Figure 50 SW FE refractivity coefficients Medium Range, Low Angle
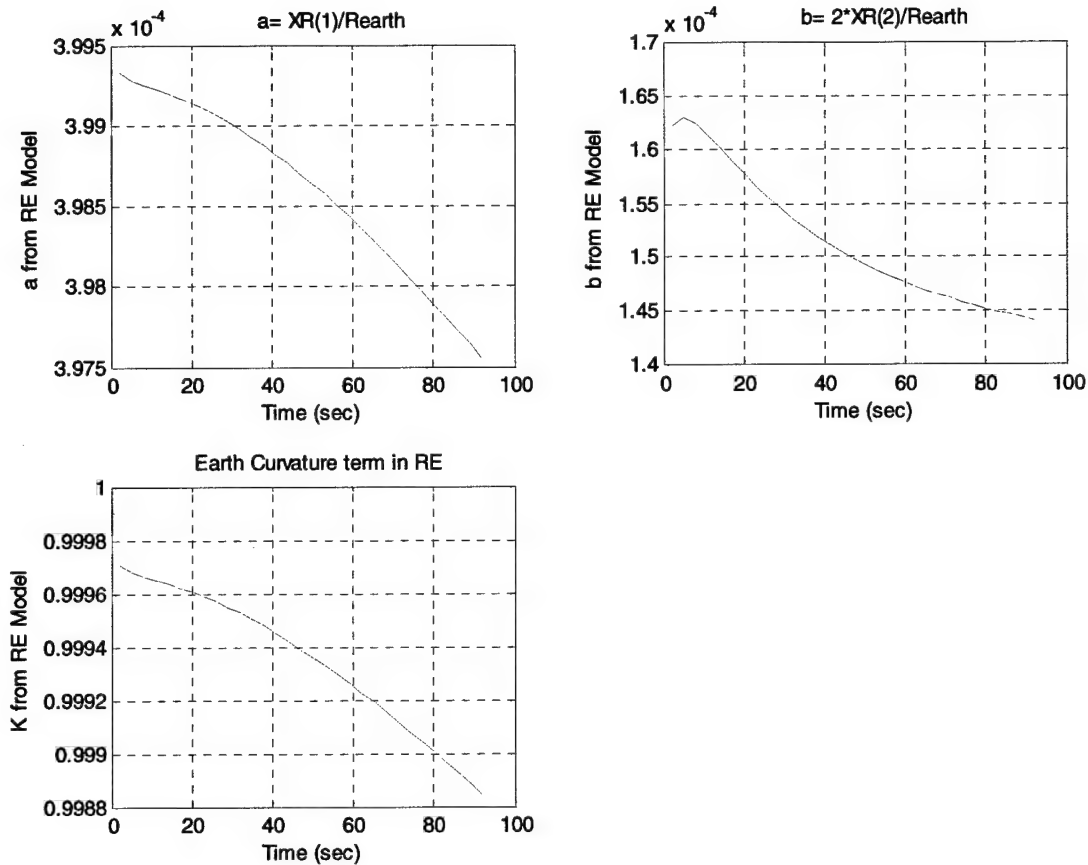
Figure 51 SW RE refractivity coefficients Medium Range, Low Angle

6.3.1.5.2 Discussion

As previously, both the Exponential and Smith-Weintraub refractivity models give virtually identical results. Both the RE and FE models attain a -4 m range error at a 60 kilometer true range, the FE takes about 50 seconds of processing to get there, while the RE is there at the beginning. The RE gives a higher refractivity, coefficient a does not converge well, but stays within a small range of the value $3.98 * 10^{-4}$, k behaves similarly around the value 0.999, coefficient b converges to a value of about $1.45 * 10^{-4}$ per meter and seems to be more observable than the other state elements. The FE refractivity coefficients seem to converge, a to about $3.495 * 10^{-4}$ and b to $4 * 10^{-5}$ per meter, this low value of b indicates a more linear refractivity.

6.3.1.6   Medium Range, Medium Angle

6.3.1.6.1 Overview

Specific Objectives

To investigate a higher elevation angle at medium true range.
Specific Conditions

Emitter- receiver: range = 60 km, elevation -10 degrees, altitude 25 km. Refer to Table
6: "Scenario Parameters" for other details.

Performance Expectations

The EXP and S-W results should be similar.

6.3.1.6.1.1 Exponential Model Results
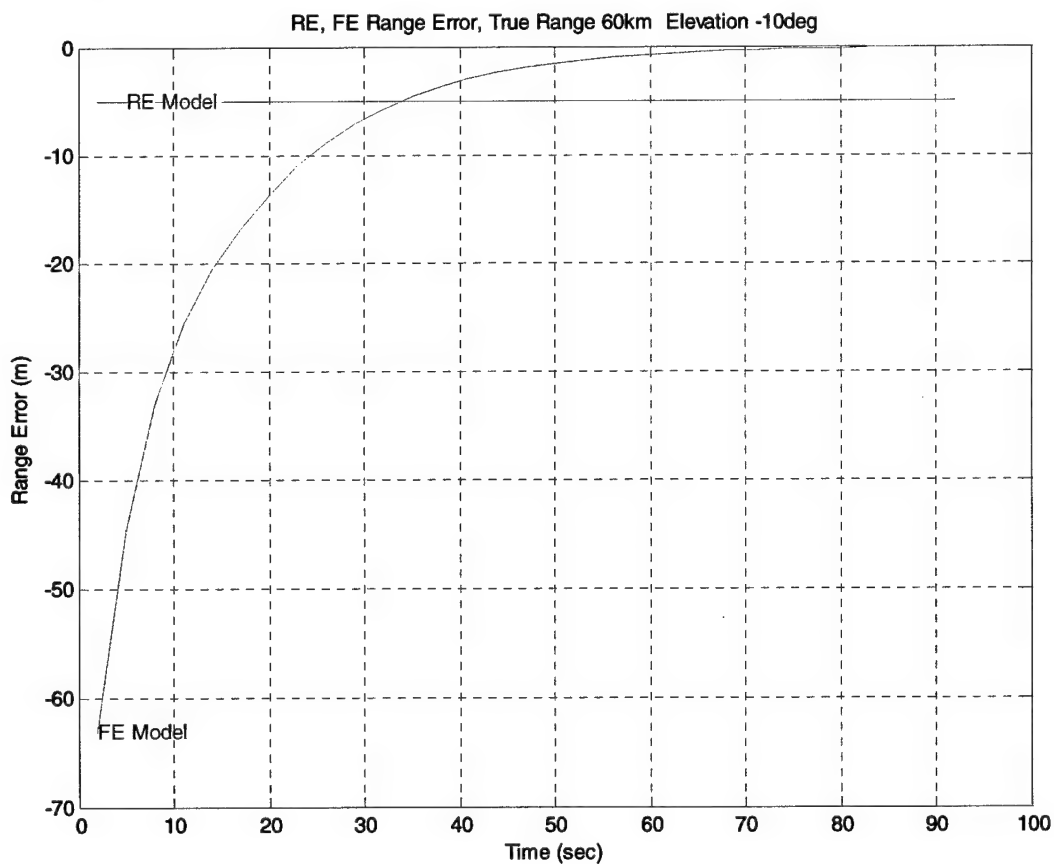
Figure 52 Medium Range, Medium Angle EXP ERRS
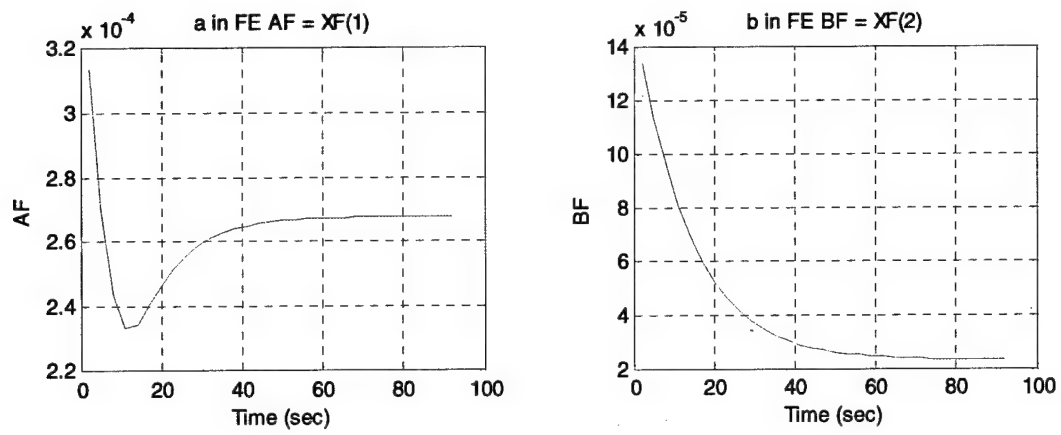
Figure 53 Medium Range, Medium Angle Exp FE refractivity coefficients
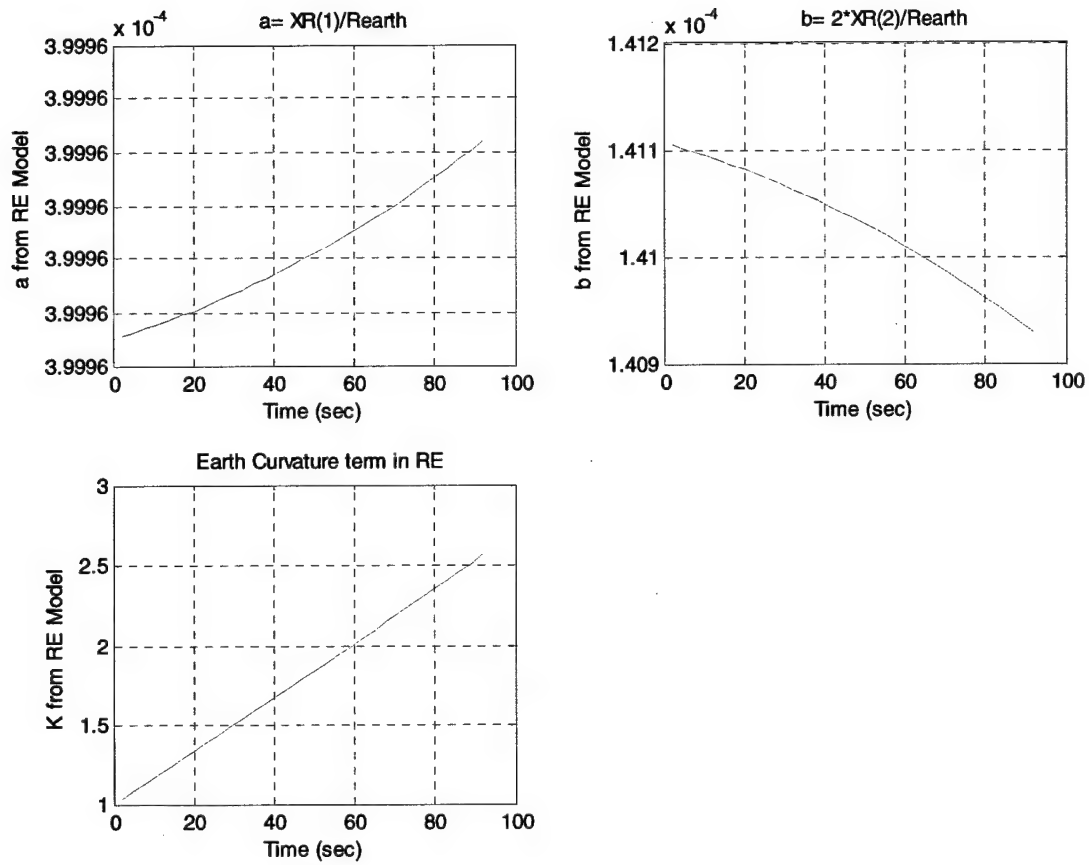
Figure 54 Medium Range, Medium Angle EXP RE refractivity coefficients

6.3.1.6.1.2 Smith-Weintraub Model Results

RE, FE Range Error, True Range 60km Elevation -10deg
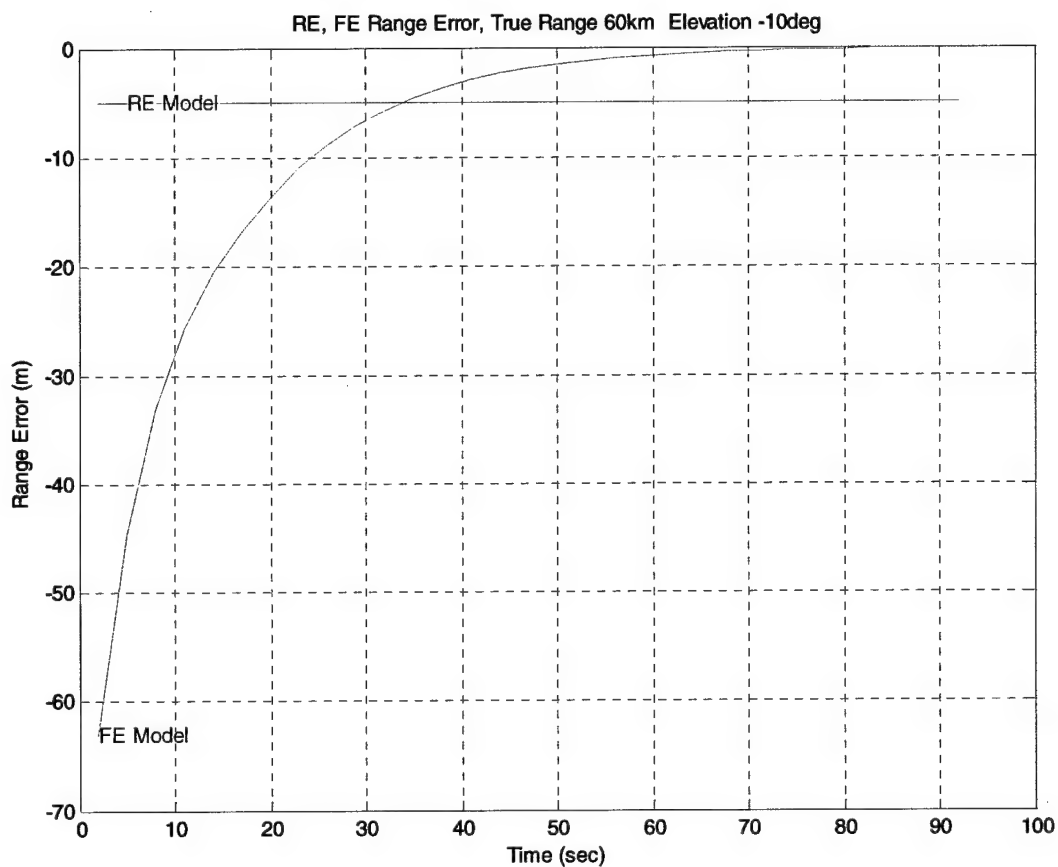
Figure 55 Medium Range, Medium Angle SW ERRS
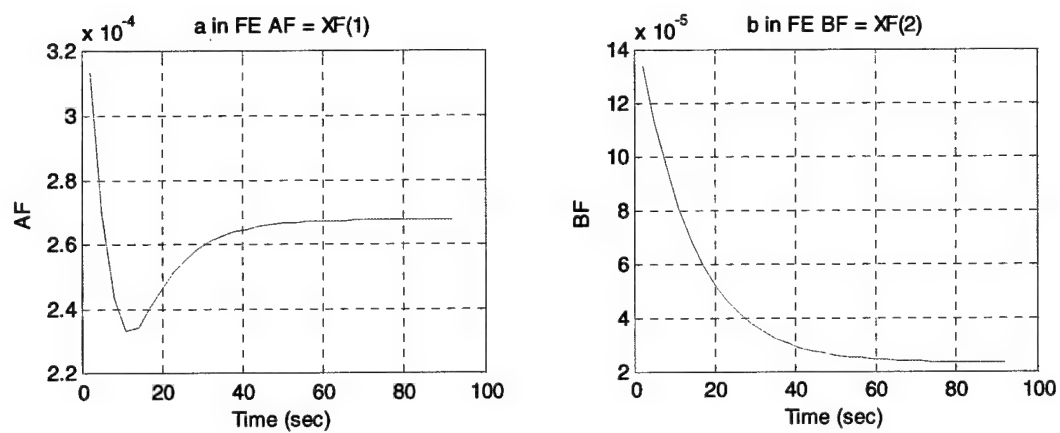
Figure 56 Medium Range, Medium Angle SW FE refractivity coefficients
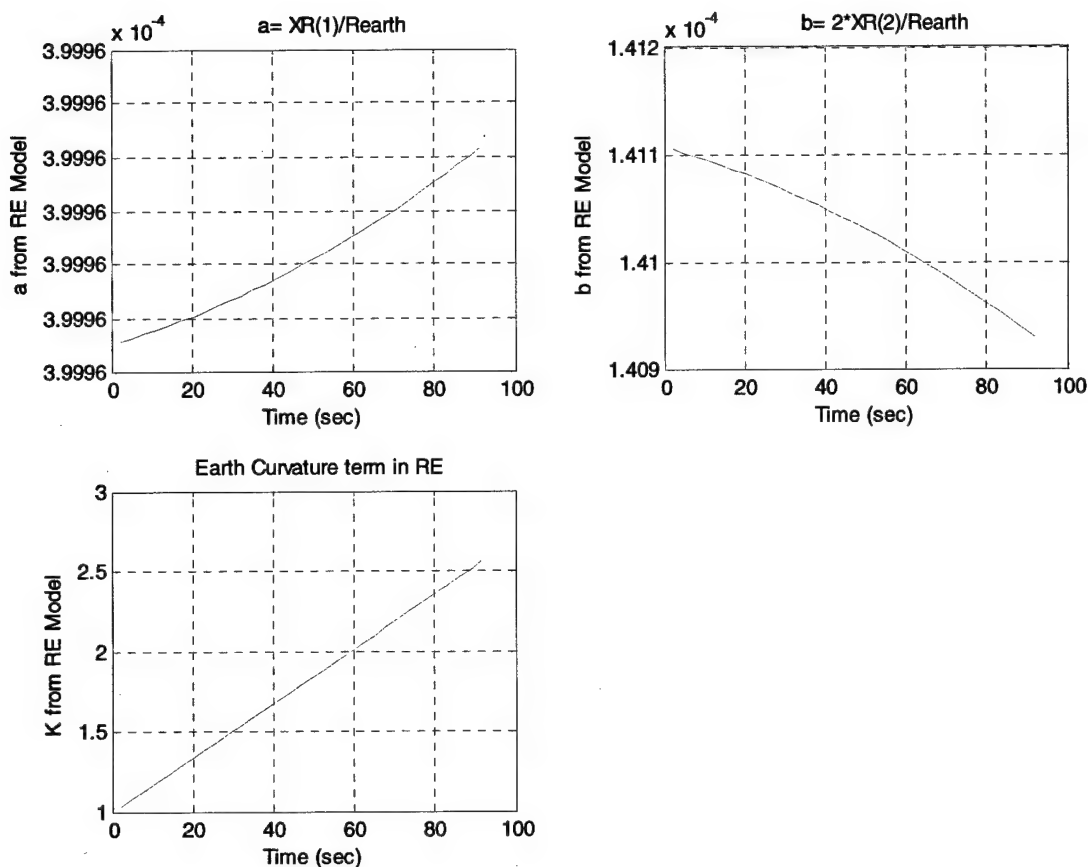
Figure 57 Medium Range, Medium Angle SW RE refractivity coefficients

6.3.1.6.2 Discussion

Again, both the Exponential and Smith-Weintraub refractivity models give virtually identical results. Here, the FE converges to a lower, nearly zero, range error, but it again takes about 50 seconds of processing to get to that value. Again, the RE is immediately at about a -4 m error. The RE gives a higher refractivity, as seen from the a term, but none of the state vector elements vary much during the processing, the curvature term, k, varies linearly during the processing, never converging, indicating that this state is not well observable. The FE result has a smaller b term of about $2 * 10^{-5}$ /m indicating a more linear fit to the refractivity and also has a small a value of about $2.6 * 10^{-4}$.

6.3.1.7   Medium Range, High Angle Case

6.3.1.7.1 Overview

Specific Objectives

To examine a medium range case at high elevation angle.

Specific Conditions

Emitter- receiver: range = 22 km, elevation 45 degrees, altitude; receiver 0 km, emitter 25 km. Refer to Table 6: "Scenario Parameters" for other details.

Performance Expectations

The EXP and S-W results should be similar, but neither the RE or FE should do well at this elevation angle.

6.3.1.7.2 Exponential Results

RE, FE Range Error, True Range 25km Elevation 45deg
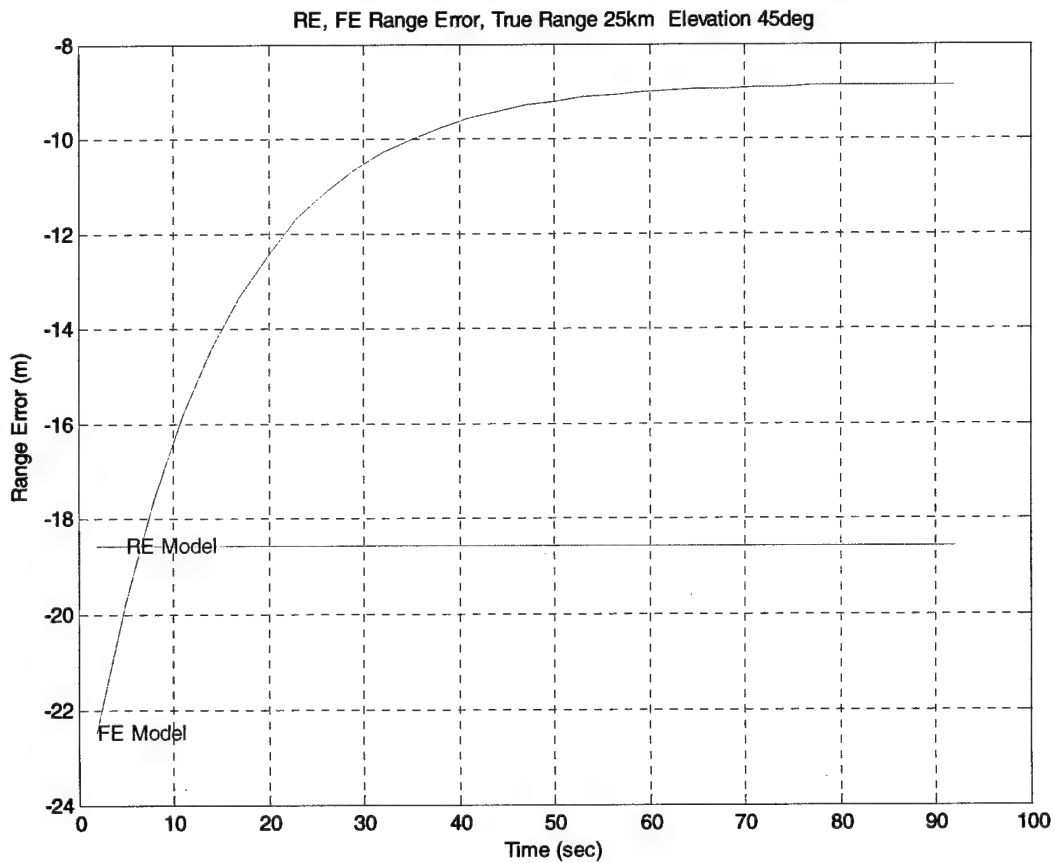
Figure 58 Medium Range, High Angle EXP ERRS
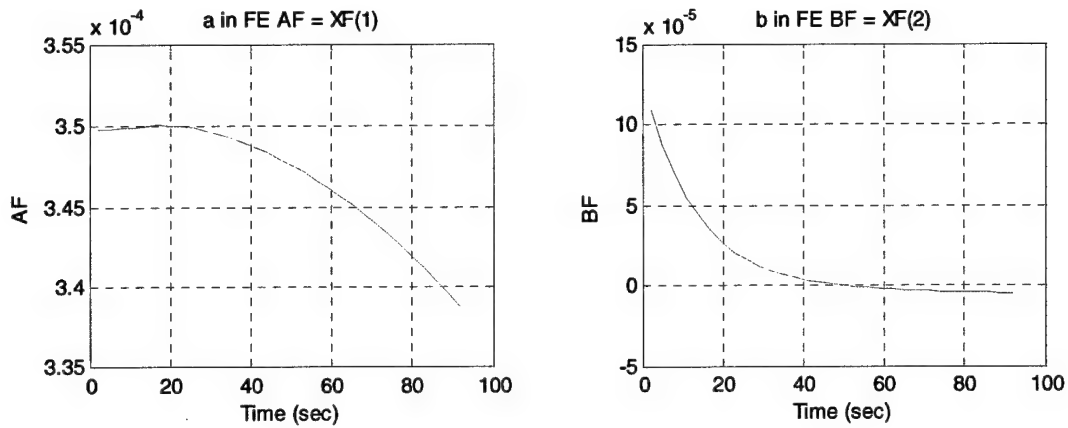
Figure 59 Medium Range, High Angle EXP FE refractivity coefficients
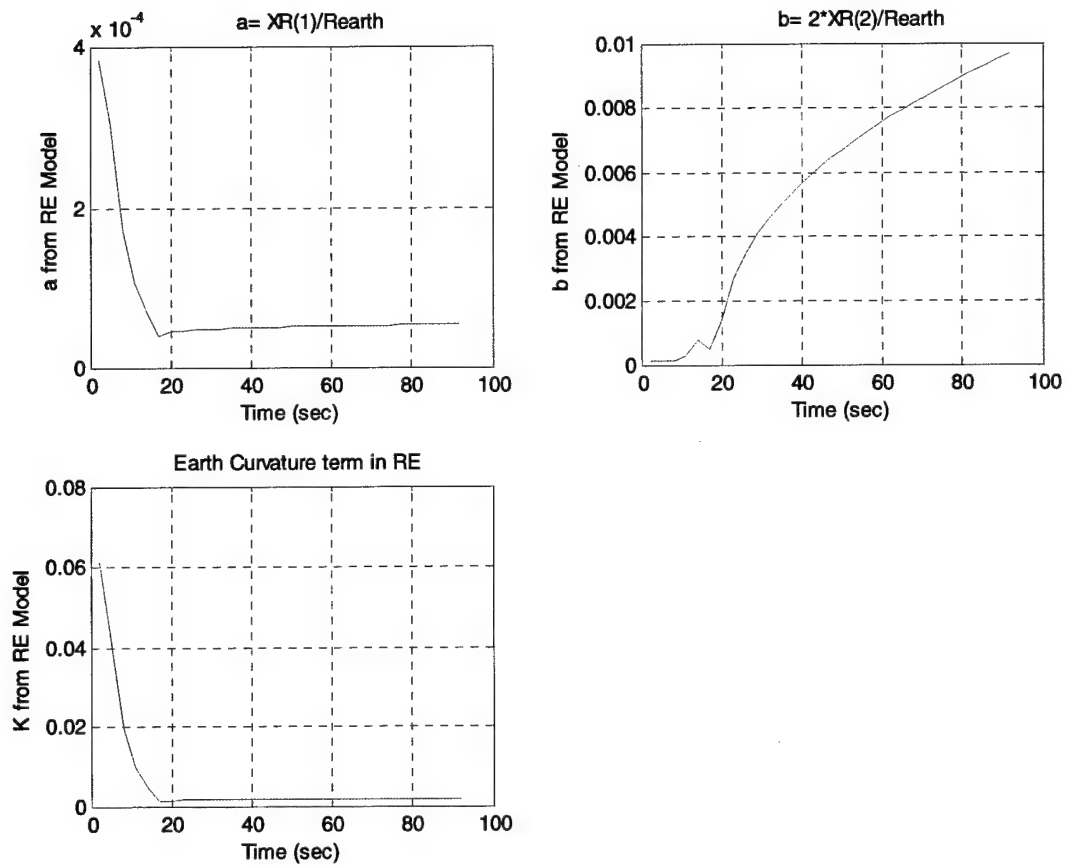
Figure 60 Medium Range, High Angle EXP RE refractivity coefficients

6.3.1.7.3 Smith-Weintraub Results
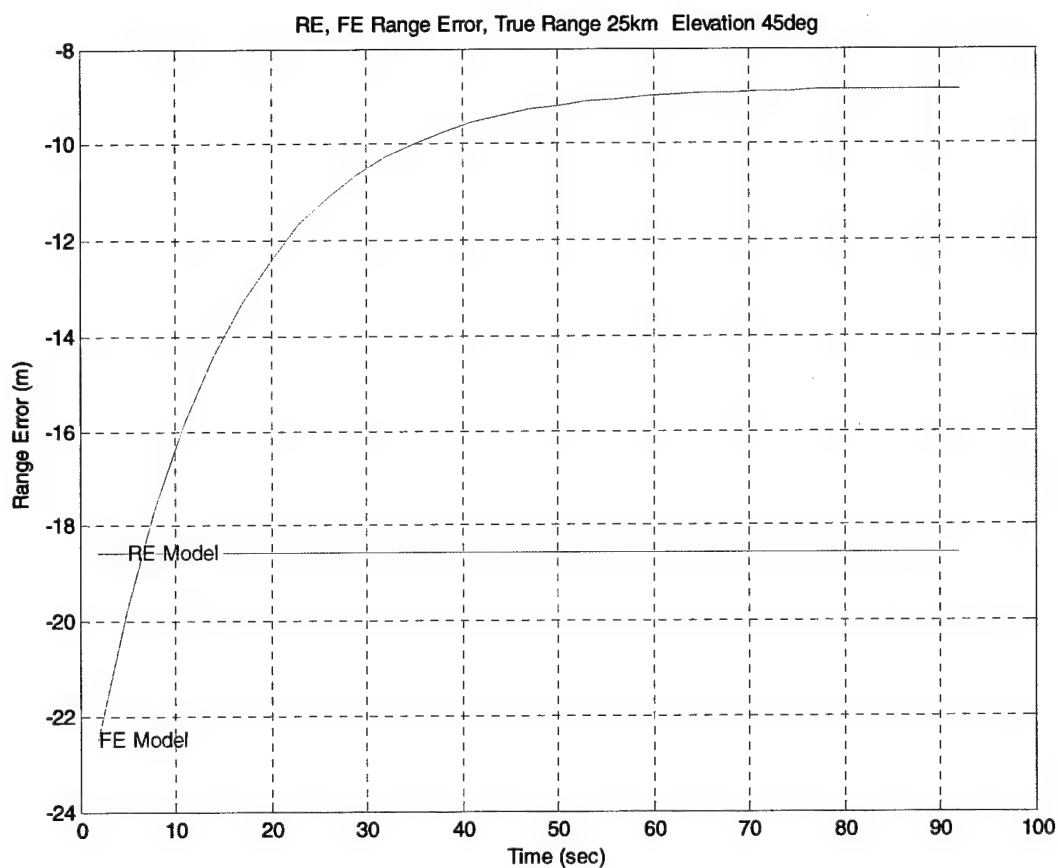
Figure 61 Medium Range, High Angle SW ERRS
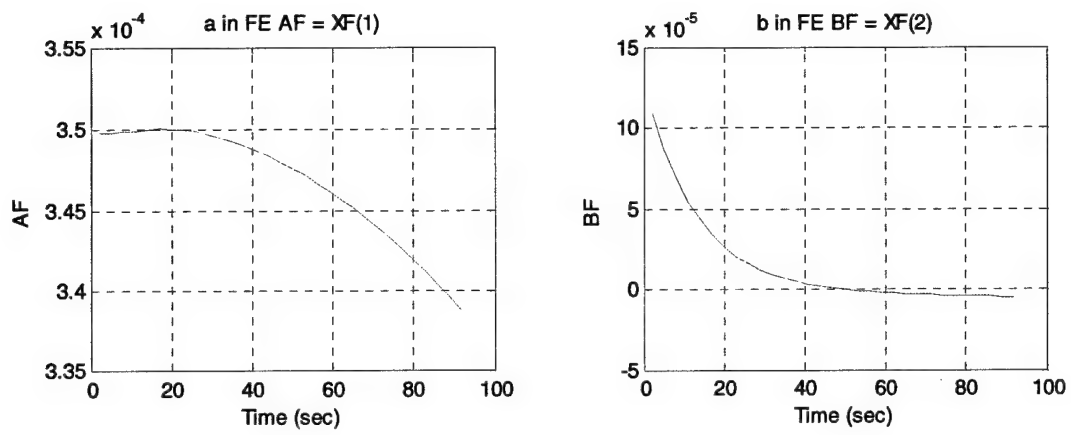
Figure 62 Medium Range, High Angle SW FE refractivity coefficients
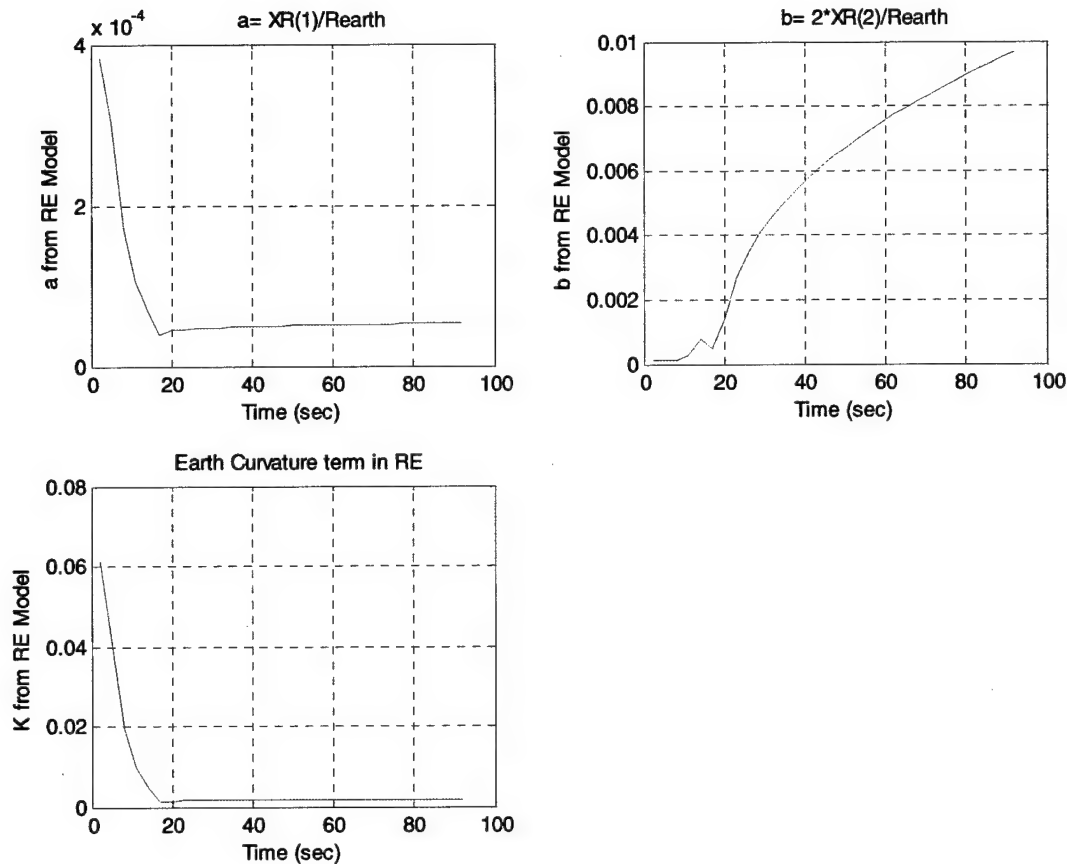
Figure 63 Medium Range, High Angle SW RE refractivity coefficients

6.3.1.7.4 Discussion

As previously, both the Exponential and Smith-Weintraub refractivity models give virtually identical results. The FE does much better than the RE here, again taking about 50 seconds to converge to about -8 m range error, while the RE is at about -18 m error throughout the processing. Not surprisingly, the RE does not give a good refractivity estimate, either, a is too low, about $5 * 10^{-5}$; b is too large at about .01 /m and k too small, nearly zero, at about 0.002. The FE refractivity is a nearly linear fit, with b at about 0/m and a about 3.4, but not settled at that value after 90 seconds of processing.

6.3.1.8   Long Range, Low Angle Case

6.3.1.8.1 Overview

Specific Objectives

To examine the long range, low emitter/receiver elevation angle case.

Specific Conditions

Emitter- receiver: range = 145 km, elevation 0.5 degrees, altitude receiver, 9 km, emitter, 12 km . Refer to Table 6: "Scenario Parameters" for other details.

Performance Expectations

The EXP and S-W results should be similar.

6.3.1.8.1.1 Exponential Model Results
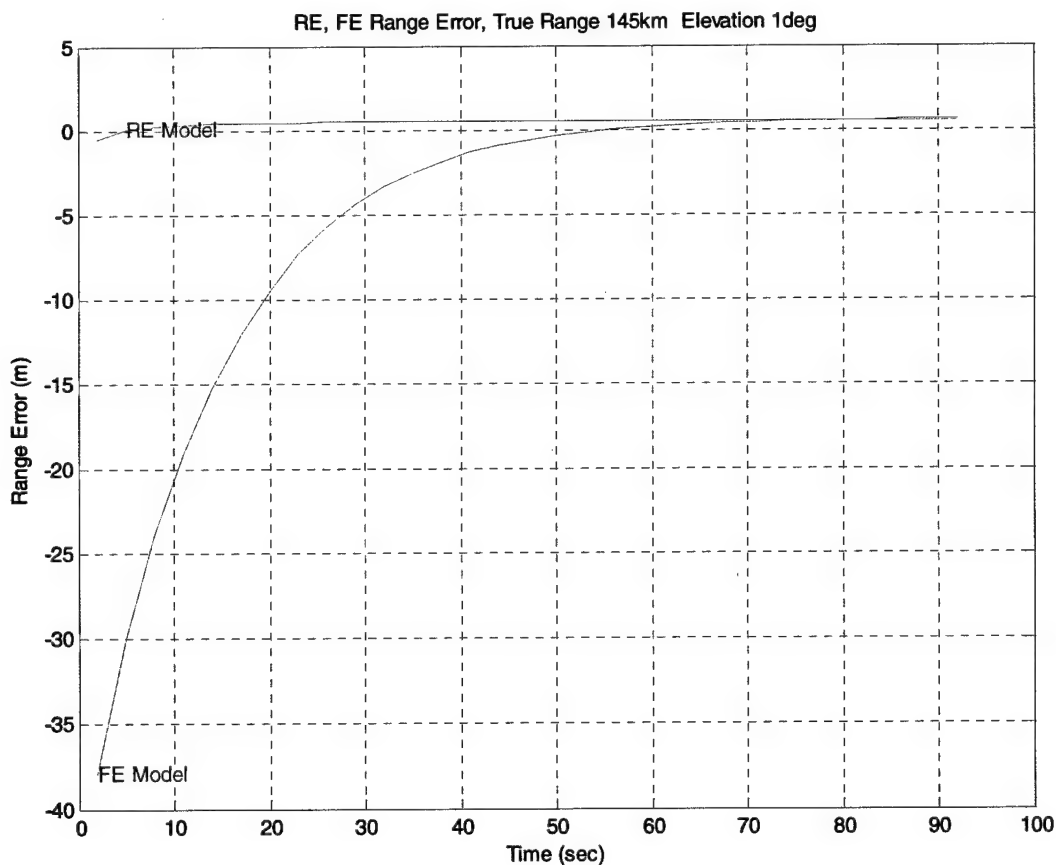
Figure 64 EXP ERRS Long Range, Low Angle
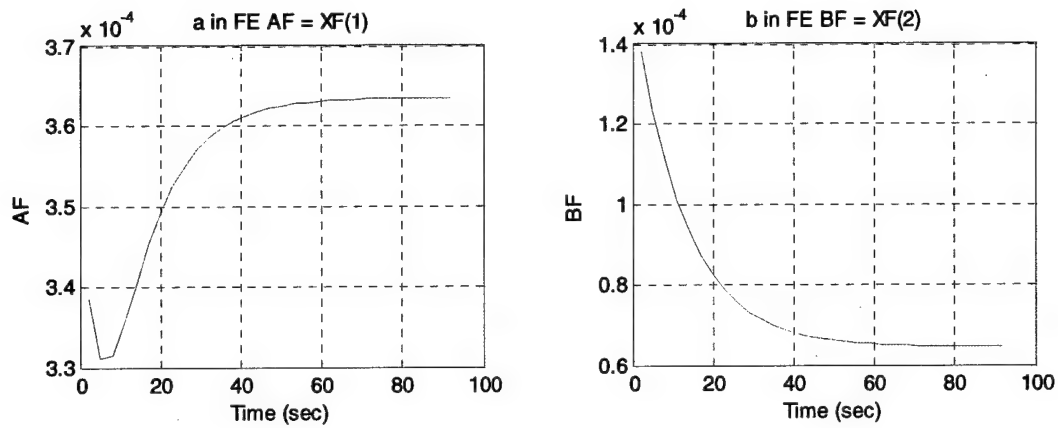
Figure 65 Long Range, Low Angle EXP FE refractivity coefficients
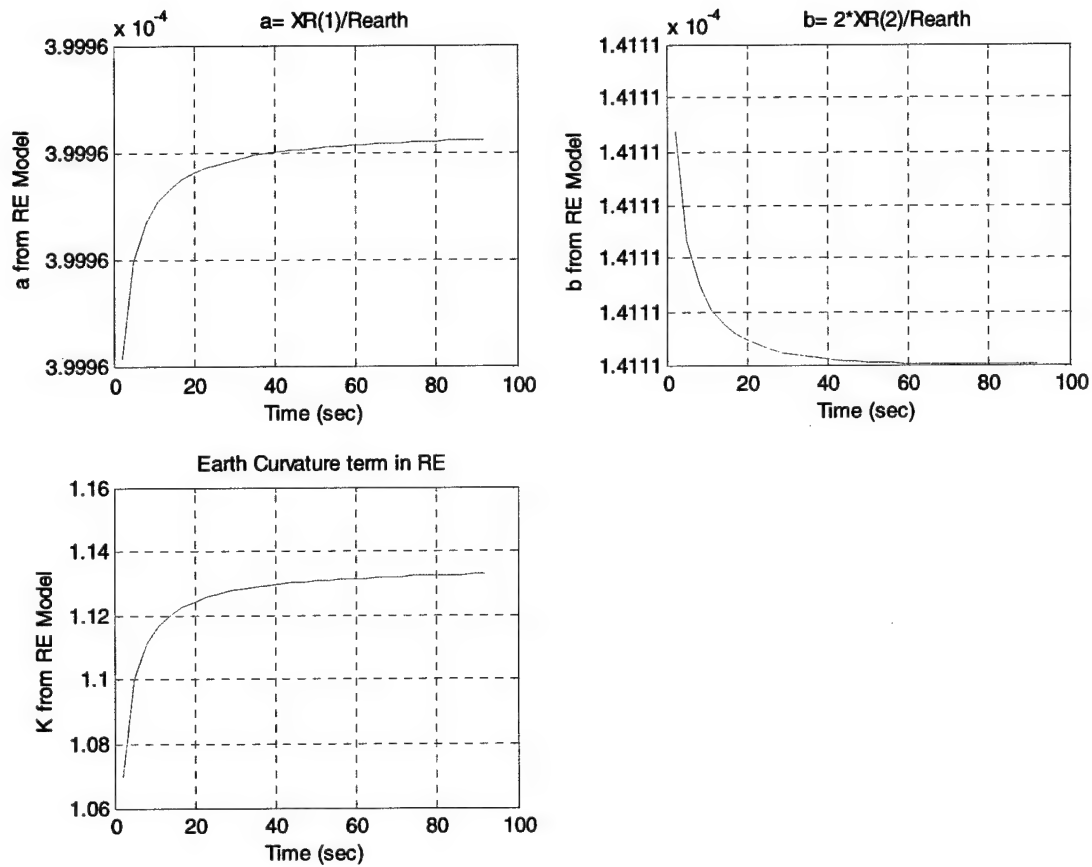
Figure 66 EXP RE refractivity coefficients Long Range, Low Angle

6.3.1.8.1.2 Smith-Weintraub Model Results

RE, FE Range Error, True Range 145km Elevation 1deg
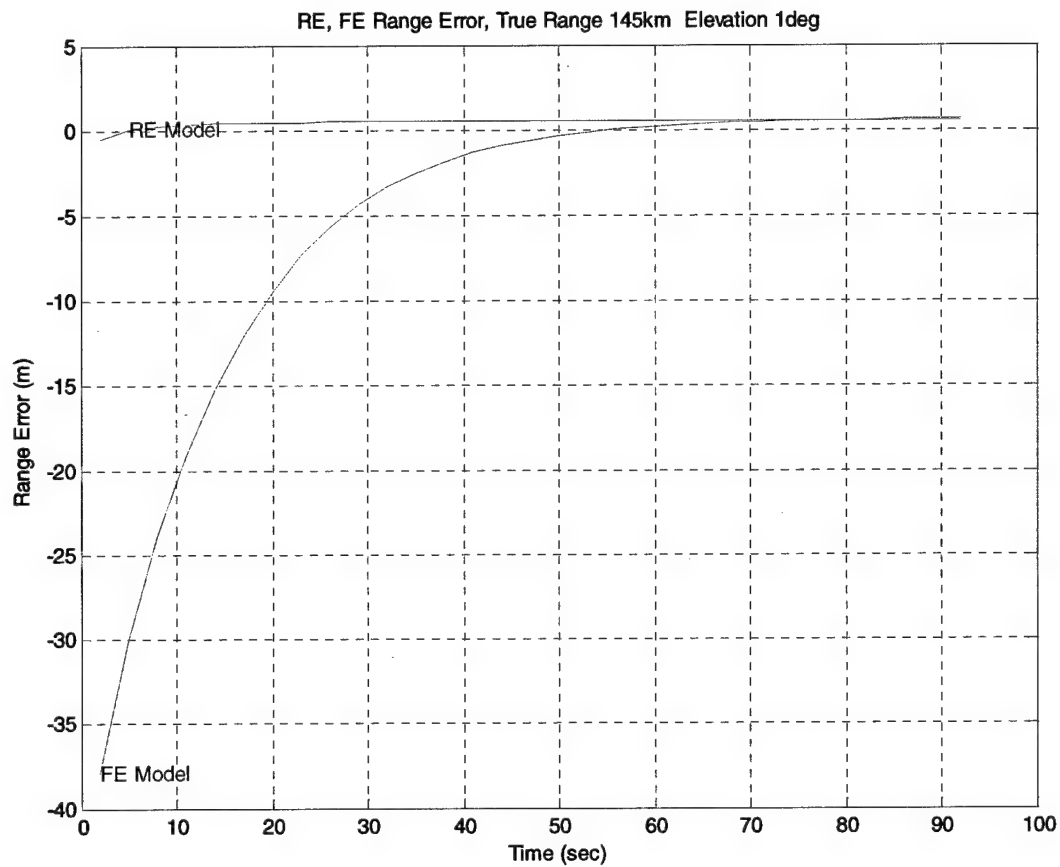
Figure 67 SW ERRS Long Range, Low Angle
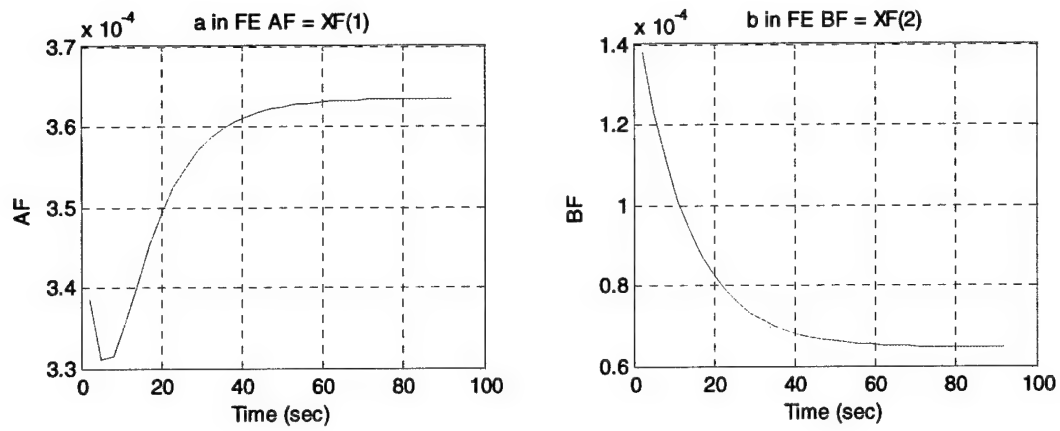
Figure 68 SW FE refractivity coefficients Long Range, Low Angle
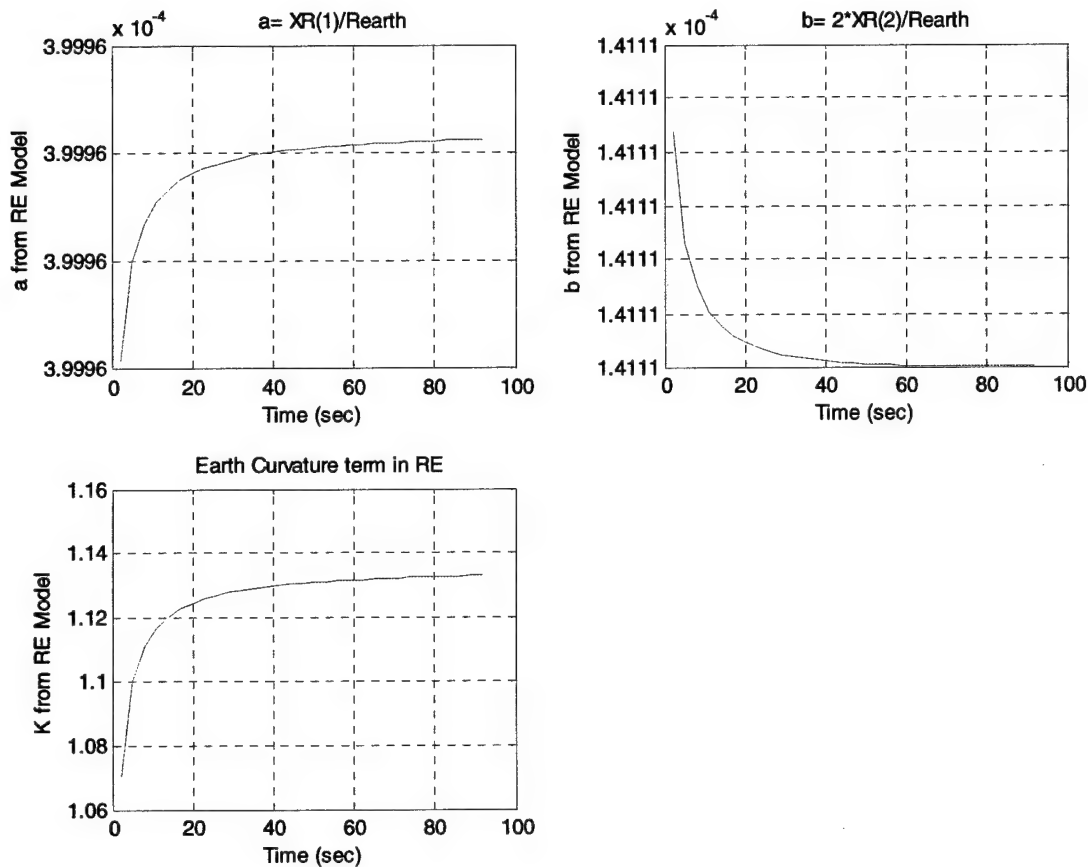
Figure 69 SW RE refractivity coefficients Long Range, Low Angle

6.3.1.8.2 Discussion

Yet again, both the Exponential and Smith-Weintraub refractivity models give virtually identical results. Both the RE and FE models do well here, attaining a low range error of less than 1 meter, although the RE converges much faster than the FE and again gives a higher refractivity estimate. The FE was not expected to do this well at this range, but proper selection of the initial covariances helped matters here.

The FE state vector elements again indicate a more linear fit to the refractivity, with b at about $7 * 10 -5$ /m, a converges to $3.65 * 10 -4$. The RE state results are a at about $4. * 10 - 4$ and b at about $1.41 * 10 -4$ /m and k converging to about 1.14 in about 40 seconds.

6.3.1.9   Long Range, Higher Angle Case

6.3.1.9.1 Overview

Specific Objectives

To examine the long range case at a higher elevation angle.

Specific Conditions

Emitter- receiver: range = 225 km, elevation 10 degrees, altitudes: receiver 1 km emitter 45 km, about as large a relative positioning as is possible. Refer to Table 6: "Scenario Parameters" for other details.

Performance Expectations

The EXP and S-W results should be similar.

6.3.1.9.1.1 Exponential Model Results
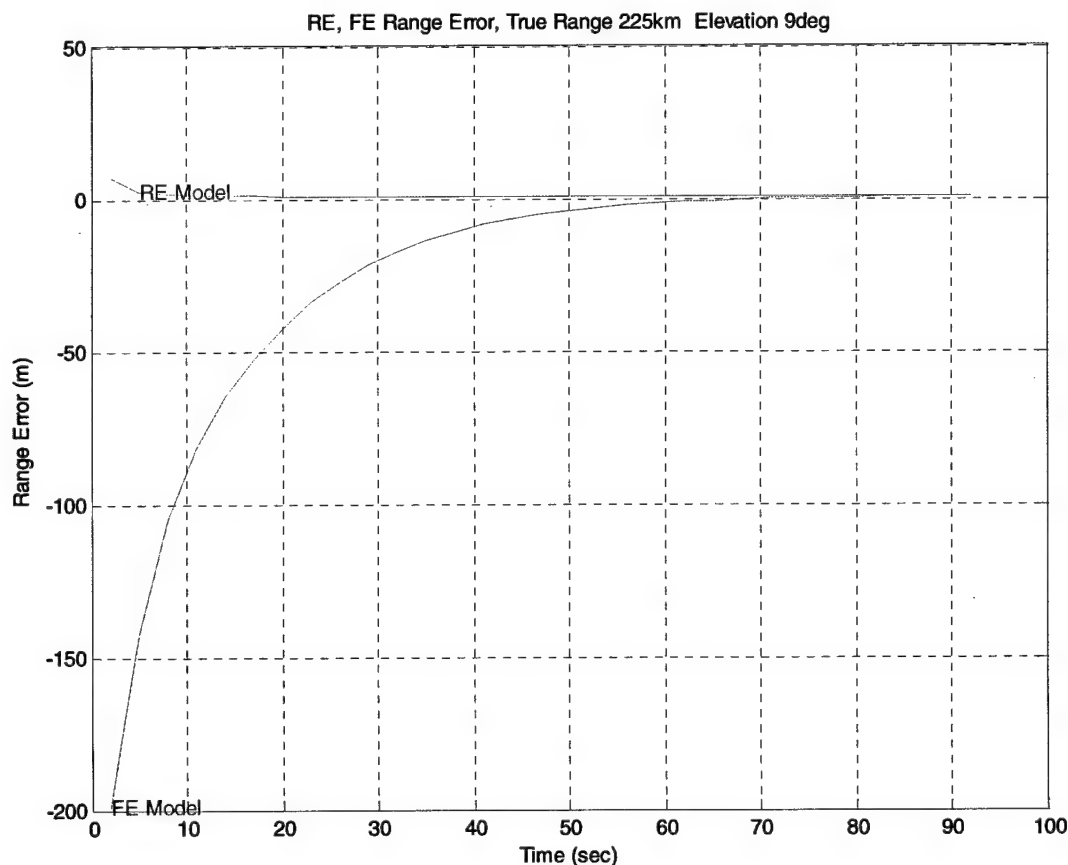
Figure 70 EXP ERRS Long Range, Higher Angle
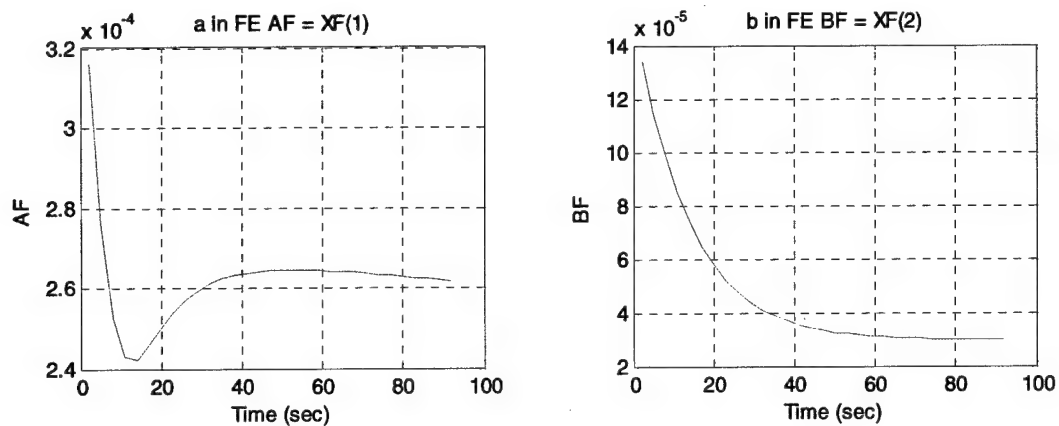
Figure 71 EXP FE refractivity coefficients Long Range, Higher Angle
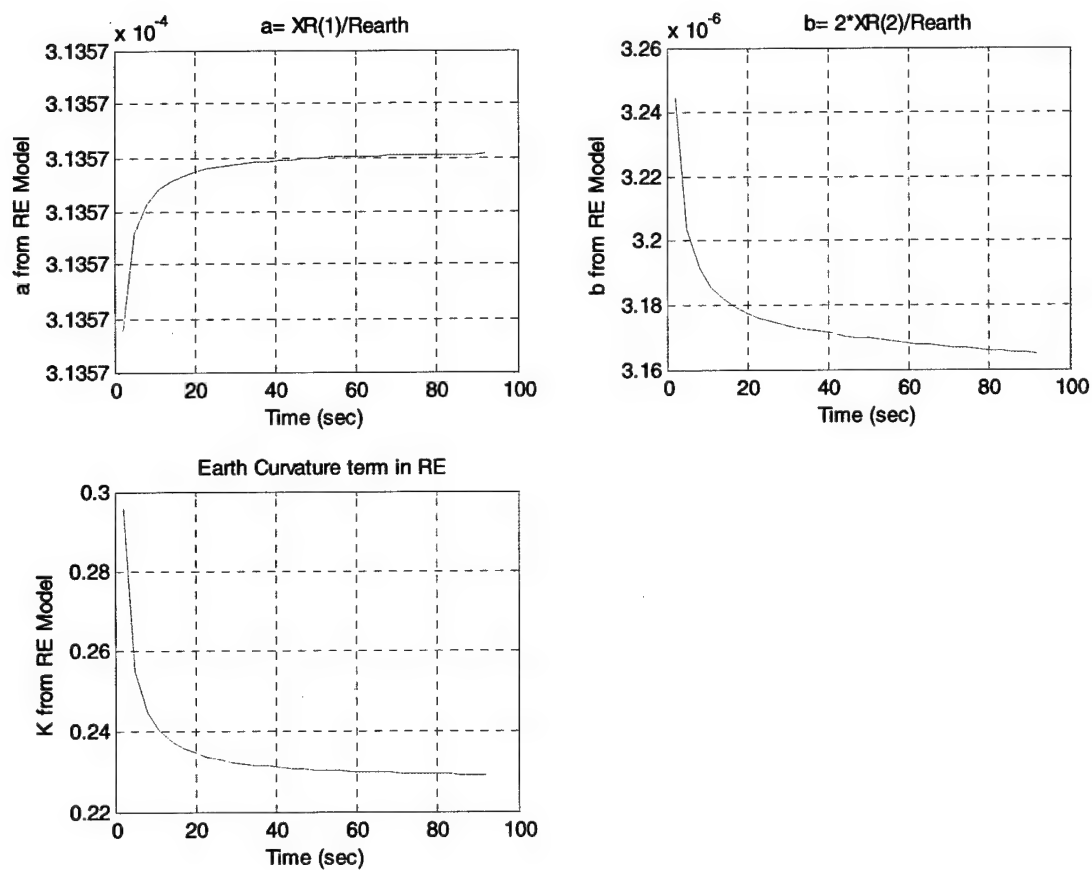
Figure 72 EXP RE refractivity coefficients Long Range, Higher Angle

6.3.1.9.1.2 Smith-Weintraub Model Results

RE, FE Range Error, True Range 225km   Elevation 9deg
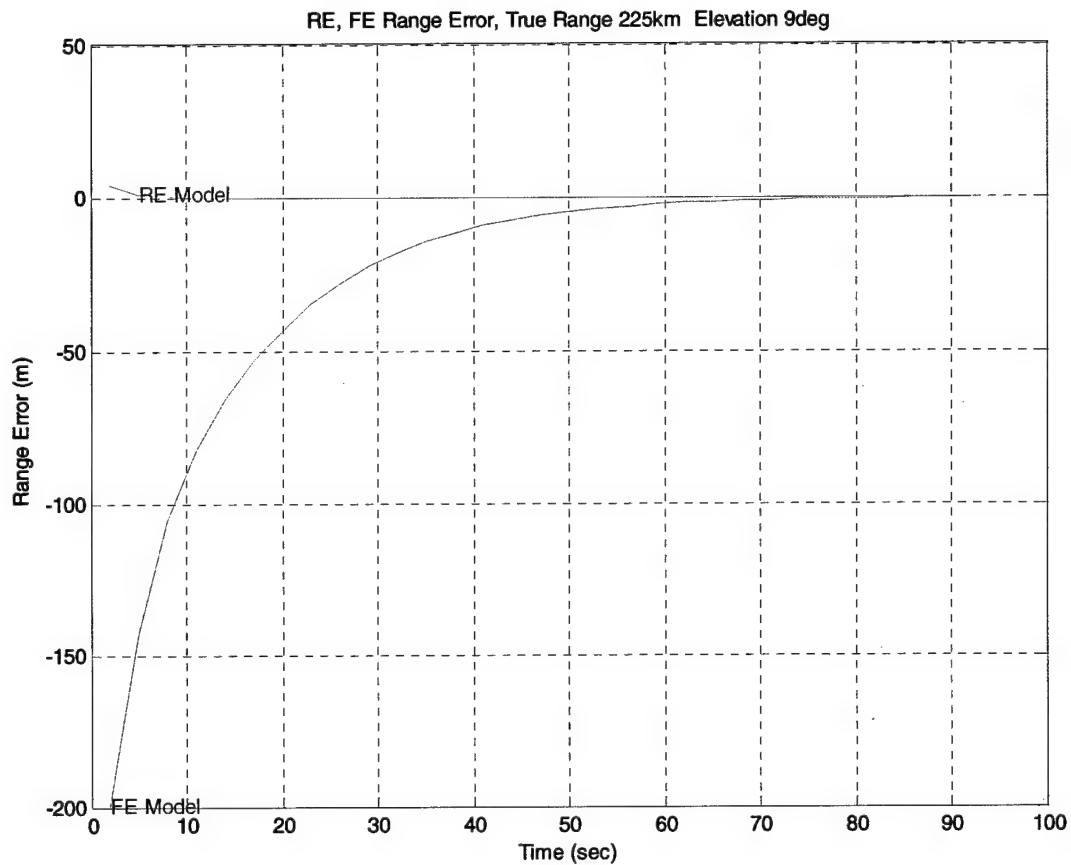
Figure 73 SW ERRS Long Range, Higher Angle
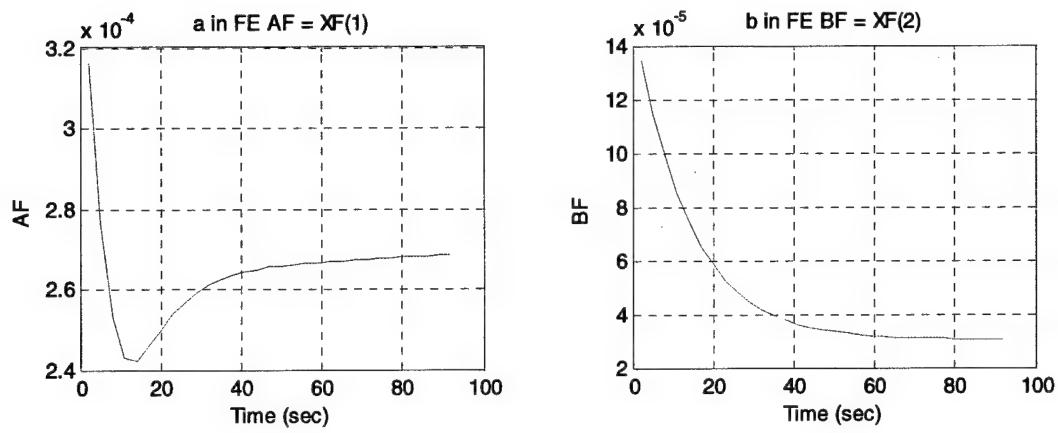
Figure 74 SW FE refractivity coefficients Long Range, Higher Angle
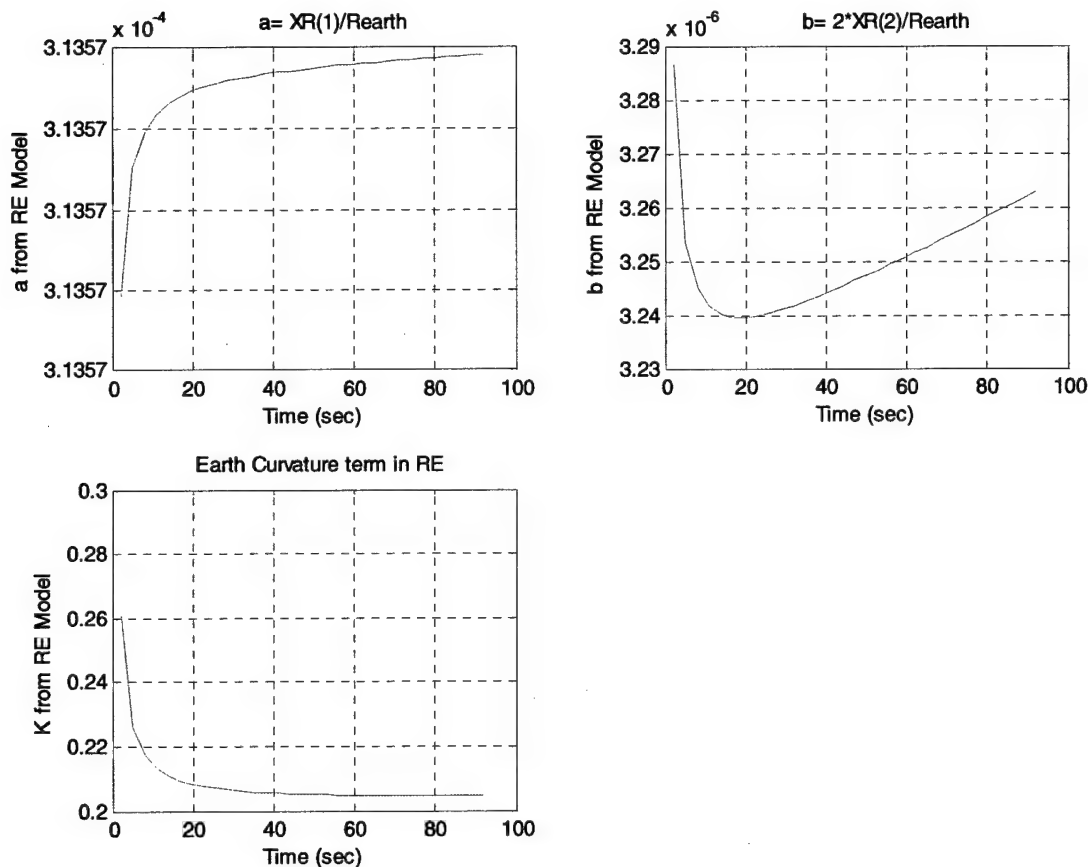
Figure 75 SW RE refractivity coefficients Long Range, Higher Angle

6.3.1.9.2 Discussion

Yet again, both the Exponential and Smith-Weintraub refractivity models give virtually identical results. Both the RE and FE models do well here, attaining a low range error of about 1 meter, although the RE converges much faster than the FE and again gives a higher refractivity estimate. The FE was not expected to do this well at this range, but proper selection of the initial covariances helped matters here.

Both model s converge as previously to a very good range estimate, but the RE gives a more reasonable refractivity estimate. The FE state vector elements again indicate a more linear fit to the refractivity, with b at about $1.0 * 10^{-5}$ /m, a converges to $2.6 * 10^{-4}$. The RE state results are a at about $3.136 * 10^{-4}$ and b at about $3.17 * 10^{-4}$ /m and k converging to a small value of about 0.23 in about 40 seconds.

6.3.1.10   Very Long Range Results

6.3.1.10.1   Overview

Specific Objectives

To examine the very long range case. The range is so long that the only feasible elevation angle to examine is a low one.

Specific Conditions

Emitter- receiver: range = 475 km, elevation 3 degrees and altitudes, receiver 25 km, emitter 45 km. Refer to Table 6: "Scenario Parameters" for other details.

Performance Expectations

The EXP and S-W results should be similar.

For comparison, the Link 16 error in this case is -240 meters.

6.3.1.10.1.1 Exponential Model Results

RE, FE Range Error, True Range 475km Elevation 3deg
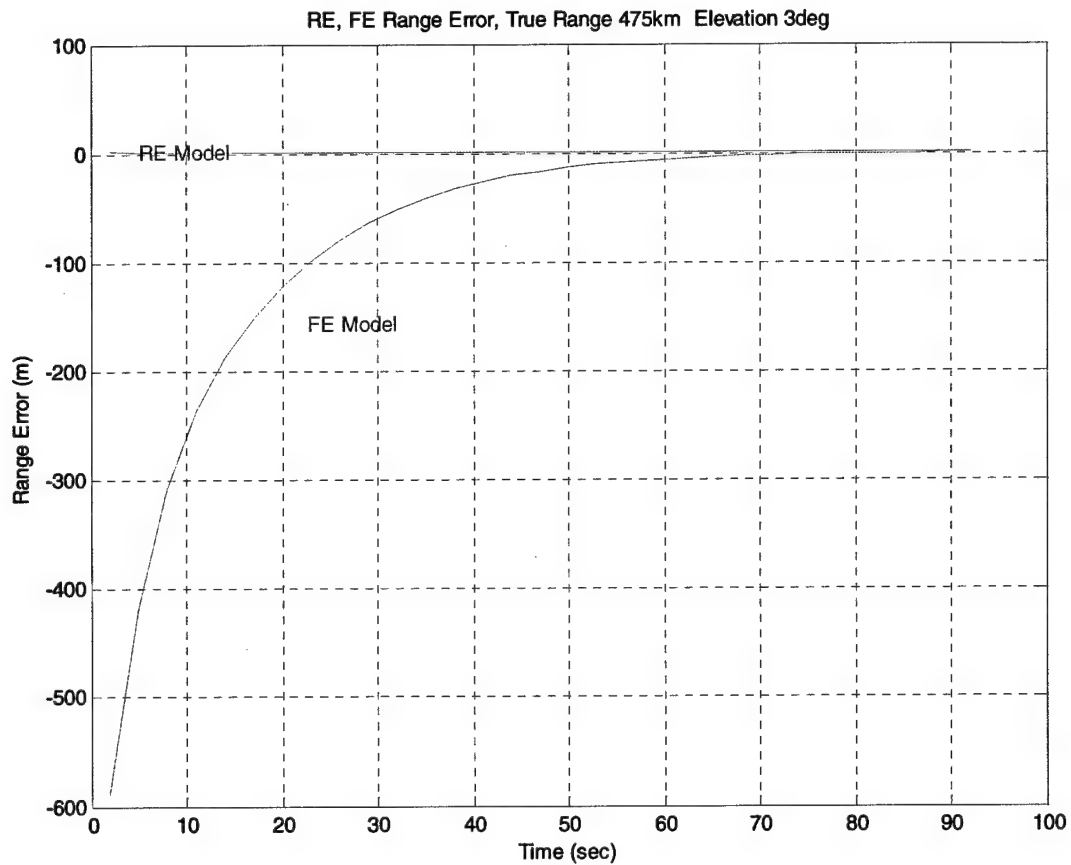
Figure 76 EXP Range Error Very Long Range Results
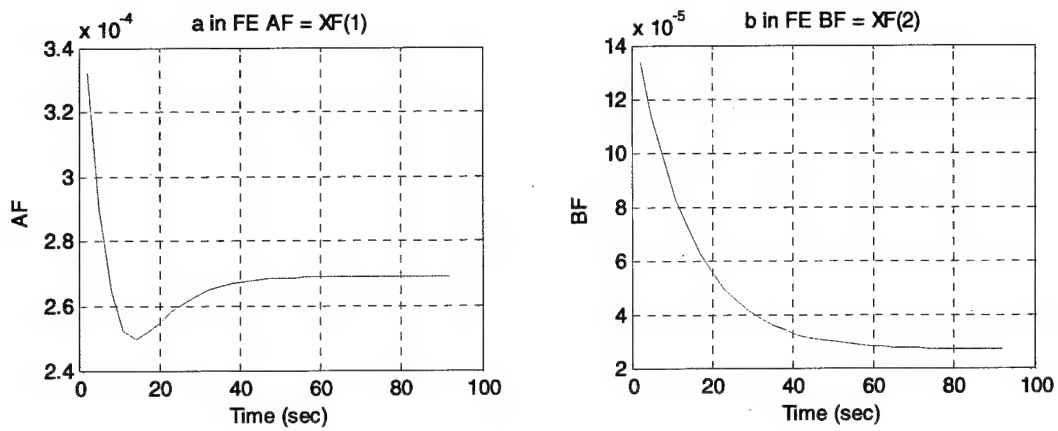
Figure 77 EXP FE refractivity coefficients Longer Range Results
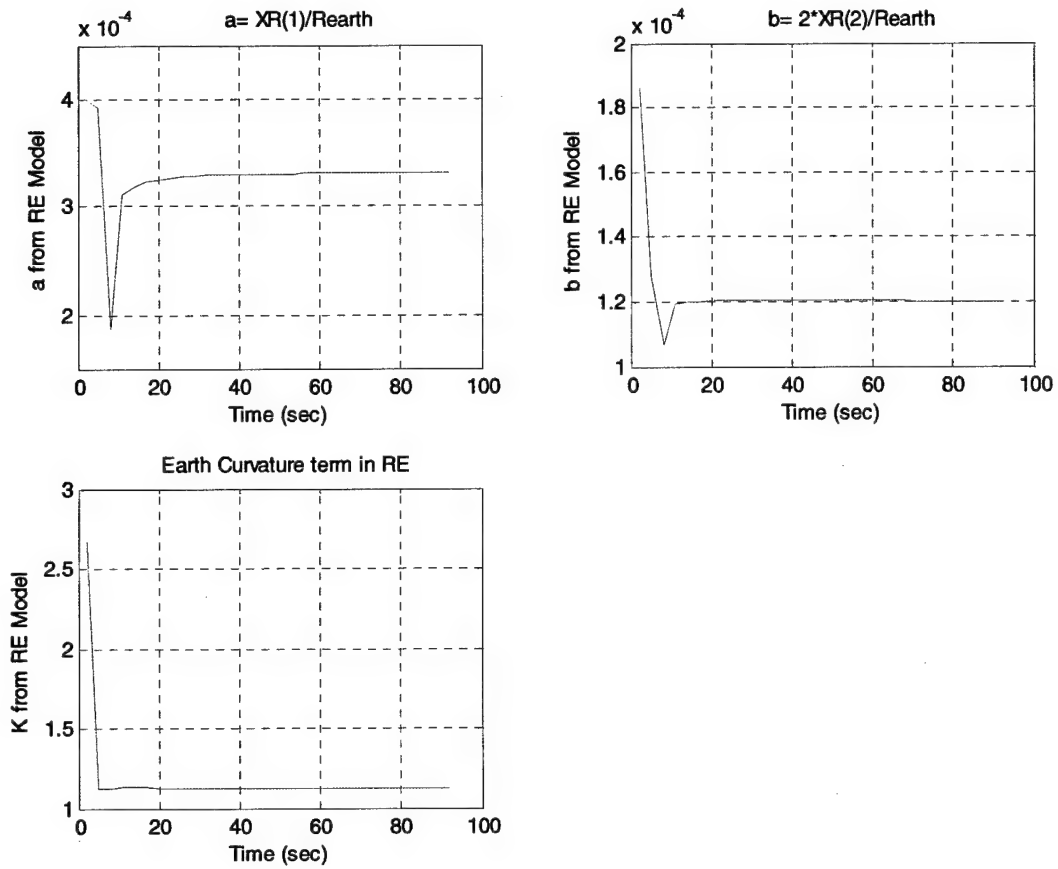
Figure 78 EXP RE refractivity coefficients Longer Range Results

6.3.1.10.1.2 Smith-Weintraub Model Results

RE, FE Range Error, True Range 475km   Elevation 3deg
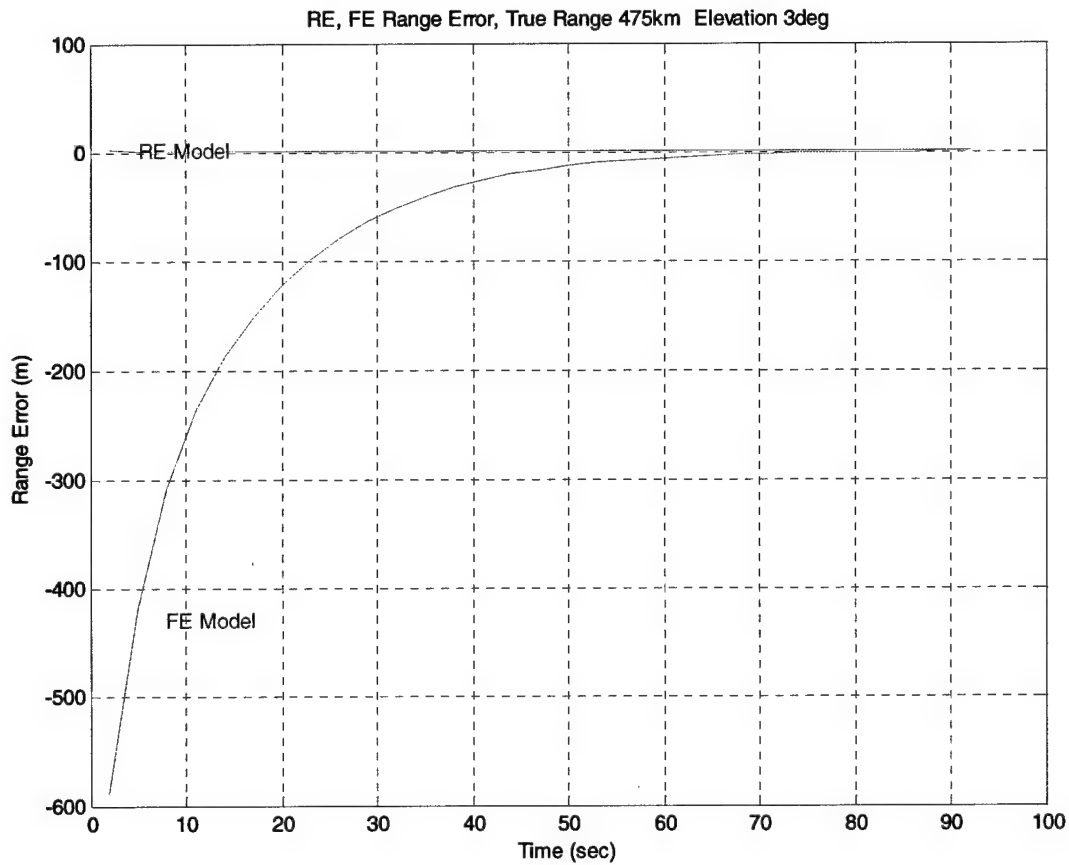
Figure 79 SW ERRS Longer Range Results
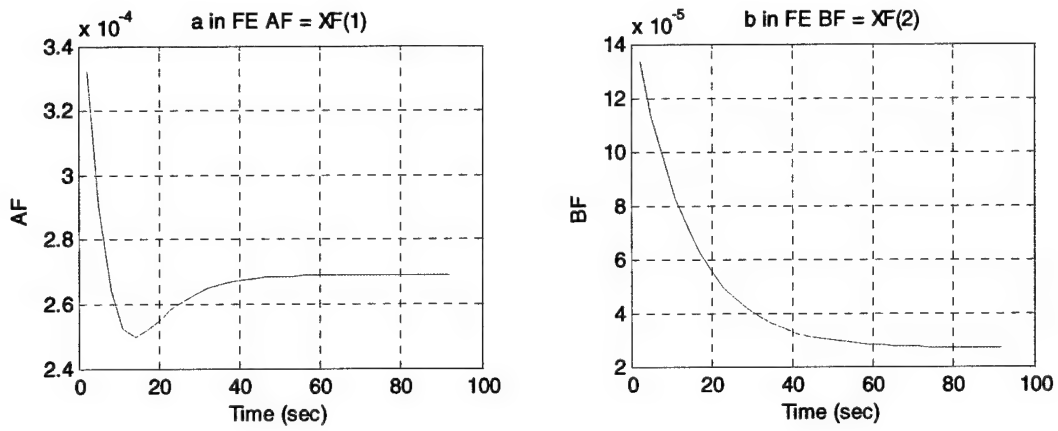
Figure 80 SW FE refractivity coefficients Longer Range Results
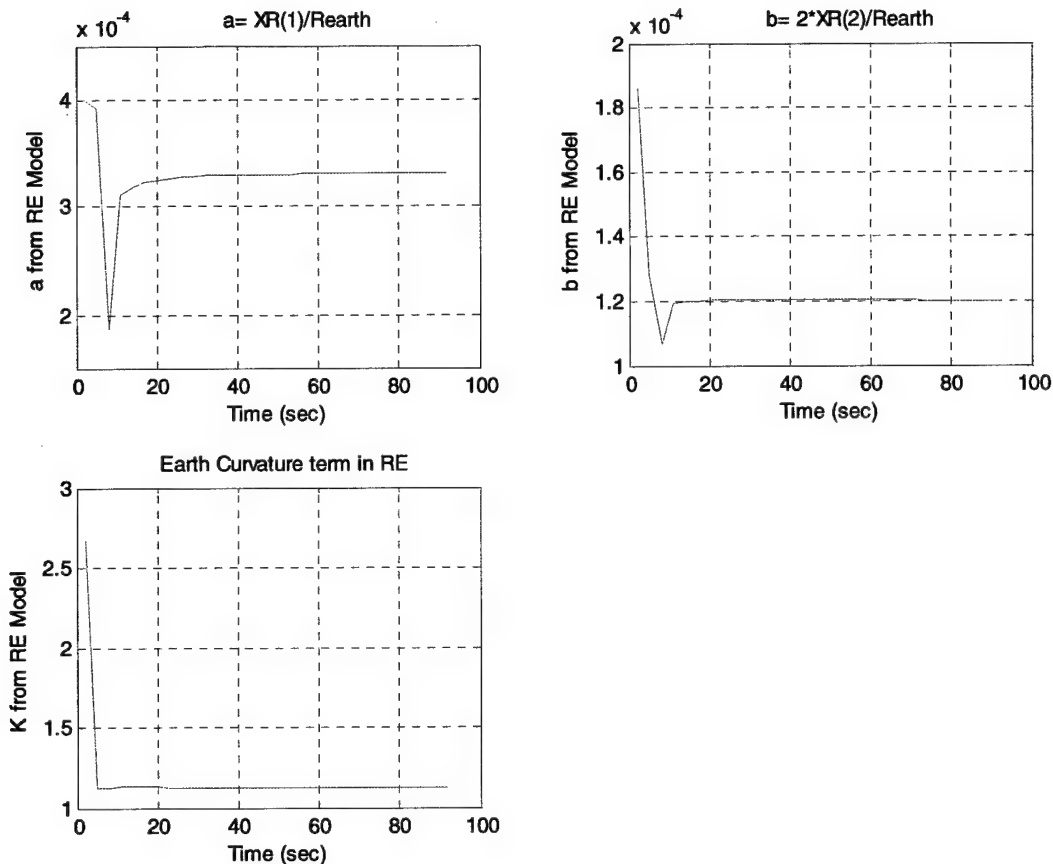
Figure 81 SW RE refractivity coefficients Very Long Range Results

6.3.1.10.2   Discussion

Yet again, both the Exponential and Smith-Weintraub refractivity models give virtually identical results.

The RE does well, arriving quickly at a small range error of about one meter rapidly, but the FE eventually converges to it, as well, surprisingly at this large a range, but only after 70 seconds of processing. The FE was not expected to do this well at this range, but proper selection of the initial covariances helped matters here, as in the long range case.   For reference, the Link-16 range error in this case, as currently implemented operationally, is about -240 meters, so either of the current models is vastly superior to it.

While both model s converge as previously to a very good range estimate, but the RE gives a more reasonable refractivity estimate. The FE state vector elements again indicate a more linear fit to the refractivity, with b at about $3.0 * 10^{-5}$ /m, a converges to $2.7 * 10^{-4}$. The RE state results are a at about $3.2 * 10^{-4}$ and b at about $1.2 * 10^{-4}$ /m, both converging in about 10 seconds while k converges to about 1.2 in about 5 seconds.

7 OVERVIEW OF RESULTS

Both the RE and FE models performed well at low elevation angles at all ranges. Surprisingly, the FE model reached smaller range error levels at long ranges at these elevations, although it took much longer for it to converge than the RE model. This is probably due to lower observabilities of the states in this model. The RE model had observability problems of a more severe nature at ranges of more than 60 km at angles larger than 10 degrees at short range and at progressively lower angles at longer ranges.

Neither model did well, as expected, at higher elevations.

These results are summarized in Table 8. Overall, it can be said that the FE model seemed to be the more robust of the two. Aside from the high emitter-receiver elevation angle case in which neither model did well, as expected, the FE gave good results, although it usually look longer than the RE. The reason for this is apparently an observability problem with RE, but it is possible that this can be corrected. This should be investigated. The measurement transformation (H) matrix in the Kalman Filter is the likely culprit, several different versions of it were implemented in an attempt to improve the situation, but none seemed to help, but this doesn't mean that it can't be improved. It involves an "error function" or erf function. It and the approximations to it used here involve an exponential of minus the elevation angle, θ_0, squared, so when the elevation is large, this exponential term can be very small, and so the diagonal elements of H would be small, thus causing it to "ignore" measurements by reducing the Kalman gain, almost having the same effect as if the measurements got noisier as the elevation angle of the signal increased.

The parameters for the RE model were adjusted more often than those of the FE model in order to optimize its performance in the various test cases. This should not be surprising as it is a more complicated model than the simpler FE model.

Test Case Scenario			Comments	FE Model performance, after 90 sec of processing	RE Model performance, after 90 sec of processing
No.	Range (km)	Elevation Angle (degrees)			
1a	9	3	RE results sensitive to initial covariance of K state.	Good	Good.
1b	7	3		slightly better than RE	Good
2	9	10		Good	Good
3	5	45		Poor	Poor
4	60	1		Good.	Good
5	60	-10		Good, but took long to converge.	Not good, refractivity parameters not observable
6	22	45	Used a different approximation to the observation matrix, HR, in RE	Fair. Insensitive to R, Q matrix values	Not good, refractivity parameters not observable
7	145	0.5		Good, but took long to converge	Good
8	225	10	RE very sensitive to R matrix values	Good, but took long to converge	Not good, refractivity parameters not observable
9	475	3	Relatively insensitive to initial conditions	Good, but took long to converge	Good

Table 7 Test Case Summary

8 RECOMMENDATIONS

8.1.1 Further Work

We recommend proceeding in developing a Link 16 operational version of the AF. To do this, the next step would be to incorporate the AF algorithm into BAE's Link 16 Navigation Simulator (LNS).

These stages of this project can be divided into two phases of the development and demonstration of the desired precision navigation algorithm.

Phase 1 – Detailed Design Study and Development of the algorithms. This is complete and is the subject of this report.

Phase 2 - The development of a full engineering test bed for the integrated operation of inertial and Link-16 navigation will be provided via a PC-based simulation tool exercising the various variants of the precision navigation algorithm.

Phase 3 – Extension of the demonstrated algorithm into actual flight hardware exercised on validated Terminal ATP assets to simulated required flight dynamics.

A laboratory environment in which is implemented a real time algorithm combing INS and Link-16 input capable of achieving community positioning accuracy of O(1) meter will be used as a testbed. The operation of an integrated INS/Link-16 system requires SELF INS data and Supporting L16 PPLI message traffic.

8.1.2 Proposed Link-16 Operational Implementation

The Atmospheric Filter would serve as a subroutine in the Link-16 Operational Computer Program (OCP). The filter must characterize atmospheric refractivity while not adding appreciably to the Link-16 OCP processing burden. There is a yet unknown limit as to how many segments of the local atmosphere in which it can characterize refractivity simultaneously *and* in real time. This is the practical limit. The physical limit would involve how best to characterize the refractivity in the locality of the home platform. That

is, in how small a volume of air does the refractivity vary to an unappreciable extent? This question may not ever be answered.

For the time being, assume that the number of regions to divide the local atmosphere is six. This number can be changed once more realistic tests are performed on the Atmospheric Filter. The FE model performs better at ranges less than 10 km at absolute elevations greater than ten degrees and the RE does better elsewhere. The atmosphere surrounding the home platform shall then be divided into six regions, three inner (short range) regions extending radially outward up to 30 km and three outer (long range) regions beyond that. One separate model will be run in each region, characterizing the refractivity therein. The FE gives better results in the inner regions, at absolute elevations of ten degrees, or higher and so will be used there and the RE is to be used at lower elevations. The RE is used in all the outer regions.

8.1.2.1 AF External Interface Characteristics

This section specifies external software interfaces for the Atmospheric Filter. The Atmospheric Filter will interface with the Link-16 Operational Computer Program (OCP) as a subroutine. It shall be called once per processing interval (i.e., once per second = 1 HZ), at which time the output from the previous processing cycle will be read by the calling routine in the OCP. The OCP will then send the routine the current processing cycle's Self Navigation and PPLI data to process.

Inputs from the Link 16 OCP (at 1 HZ):
Self Navigation Solution: SELF NAV latitude, longitude, altitude, time of validity, north velocity, east velocity, altitude rate.
Link-16 emitter PPLI data: PPLI latitude, longitude, altitude, time of validity, TOA Control input.

Outputs to the Link 16 OCP (at 1 HZ):
AF Filter Solution Results Message.
AF Status Message for a given Link-16 emitter.

Possible Operational Rules for the Application of AF Corrections to OCP Speed of Light would be:

- No AF speed of light correction for any Link-16 emitter shall be applied to the OCP until steady state operation of that emitter's AF filter has been achieved. (Filter Q > 10)

- No processing of any of the six AF filters shall be permitted unless Own (Self) Navigation is valid, and a relative quality (Q_{PR}) level of at least 10 has been achieved. Once this level has been achieved, processing in that filter will continue while the quality remains above 8.

- No data from a Link-16 emitter shall be accepted for AF processing unless it has achieved a certain relative quality level.

- The observation variance, R, used for processing of any Link-16 emitter shall be a function of that emitter's position quality.

- The speed of light correction for any Link-16 emitter not identified with a steady state AF filter shall use the existing (i.e., default) algorithm.

Refer to Appendix E for the proposed Atmospheric Filter Software to be included in the Link 16 OCP.

9 REFERENCES

1. Collins, P. and Langley, R. "Limiting Factors in Tropospheric Propagation Delay Error Modeling for GPS Airborne Navigation," Presented at the Institute if Navigation 52nd Annual Meeting, Cambridge, MA, June 19-21, 1996.
2. Flores, A., Ruffini, G. and Rius, A. "4D Tropospheric Tomography using GPS Estimated Slant Delays." Annales Geophysicae (2000) 18: 223 – 234.
3. Thayer, D., An Improved Equation for the Radio Refractive Index of Air, *Radio Sci.*, 9, 803-807, 1974.
4. J. Coleman. "Atmospheric Refraction Effects with Application to Link-16 TOA Compensation," 13 June 2000 Draft. Internal BAE document.
5. H. R. Reed & C. M. Russell, *Ultra-High Frequency Propagation*, Wiley, 1953. Chapter 3.
6. Bean, B.R. and Thayer, G.D., "Models of Atmospheric Radio Refractive Index," Proc. Of the IRE, May 1959.
7. Singer Kearfott, Internal Document Y258A002.
8. AEGIS Program Manager, *Understanding Link-16*, April 1994, Logicon, Inc., Washington, D.C., pp. 3-7.
9. Gelb, A., ed., *Applied Optimal Estimation*, The Analytical Sciences Corp., MIT Press, 1974 (Kalman Filtering reference).
10. Mendes, V.B. and Langley, R.B. "A Comprehensive Analysis of Mapping Functions Used in Modeling Tropospheric Propagation Delay in Space Geodetic Data," Presented KIS94, International Symposium on Kinematic Systems in Geodesy, Geomantics and Navigation, Banff, Canada, August 30 – September 2, 1994.
11. University of Wyoming, College of Engineering, Department of Atmospheric Sciences, Upper Air Meteorological Soundings Website. Meteorological data from hundreds of weather stations all over the world, twice a day, recent readings are available and the data goes back several years. URL: http://weather.uwyo.edu/upperair/sounding.html

10 LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DoD	Department of Defense
DTB	Data Transfer Blocks

DTI	Data Transfer Interrupt
ECR	Earth Centered Rotating
I/O	Input/Output
Init	Initialization
IRS	Interface Requirement Specification
JTIDS	Joint Tactical Information Distribution System
kbps	Kilobits Per Second
L16	Link-16
LNS	Link 16 Navigation Simulation software system
MIDS	Multifunctional Information Distribution System
Hz	Hertz
ms	milliseconds
NAV	Navigation
nm	Nautical Miles
NPS	Network Participation Status
OCP	Link-16 Operational Computer Program
OTTS	Original Transmit Time Slot
PEQ	Position Extrapolation Quality
PNC	Primary Navigation Controller
PPLI	Precise Participant Location and Identification
RELNAV	Relative Navigation
SDP	Software Development Plan
SOW	Statement of Work
SRS	Software Requirement Specification
STN	Source Track Number (a Link 16 Unit's ID)
TADIL-J	Tactical Data Information Link - JTIDS
TOA	PPLI Time of Arrival
TOV	PPLI Time of Validity
U	Unclassified

11 Appendix A Meteorological Data Used to Compare Refractivity Models

The following is the text of the meteorological data obtained from The University of Wyoming, College of Engineering, Department of Atmospheric Sciences, Upper Air Meteorological Soundings Website used in this study to compare refractivity models in Figures 3 and 4 of this report.

02527 Goteborg Observations at 12Z 04 Dec 2002

PRES	HGHT	TEMP	DWPT	RELH	MIXR	DRCT	SKNT	THTA	THTE	THTV
hPa	m	C	C	%	g/kg	deg	knot	K	K	K
1004.0	164	0.6	-0.4	93	3.72	70	1	273.4	283.7	274.1
1000.0	199	0.2	-1.4	89	3.47	85	5	273.4	282.9	273.9
988.0	295	-0.5	-1.9	90	3.38	100	11	273.6	283.0	274.2
941.0	683	-3.3	-3.9	95	3.05	115	20	274.6	283.2	275.1
937.0	717	-3.5	-4.1	96	3.02	114	20	274.7	283.2	275.2
931.0	768	-3.3	-3.6	98	3.16	112	21	275.4	284.3	275.9
927.0	802	-0.7	-1.1	97	3.82	111	22	278.4	289.1	279.1
925.0	819	-0.7	-1.2	96	3.80	110	22	278.6	289.3	279.2
921.0	854	-0.7	-2.7	86	3.42	108	22	278.9	288.6	279.5
917.0	888	-0.9	-3.0	85	3.35	105	22	279.1	288.6	279.6
850.0	1491	-4.3	-9.1	69	2.27	90	14	281.6	288.3	282.0
846.0	1528	-4.5	-9.2	70	2.26	89	14	281.8	288.4	282.2
779.0	2169	-9.8	-11.7	86	2.02	65	11	282.8	288.8	283.1
752.0	2442	-12.1	-12.7	95	1.92	87	9	283.2	288.9	283.5
748.0	2483	-11.5	-13.3	87	1.84	90	9	284.3	289.8	284.6
746.0	2504	-10.7	-18.7	52	1.18	95	9	285.4	289.0	285.6
736.0	2608	-7.5	-21.5	32	0.94	120	10	290.0	293.0	290.1
724.0	2736	-7.1	-23.7	25	0.79	150	11	291.8	294.4	291.9
713.0	2856	-6.7	-25.7	20	0.67	160	3	293.5	295.7	293.6
700.0	2999	-7.3	-30.3	14	0.44	170	11	294.4	295.9	294.4
695.0	3055	-7.5	-32.5	12	0.36	176	13	294.8	296.0	294.8
691.0	3100	-7.7	-32.6	12	0.36	180	15	295.0	296.2	295.1
666.0	3385	-9.2	-33.2	12	0.35	195	11	296.5	297.7	296.5
664.0	3409	-9.3	-33.3	12	0.35	194	11	296.6	297.8	296.7
659.0	3467	-9.5	-22.5	34	0.96	193	12	297.0	300.2	297.2
647.0	3609	-10.3	-36.3	10	0.27	188	12	297.7	298.6	297.7
639.0	3704	-11.1	-31.1	17	0.45	185	13	297.8	299.4	297.9
633.0	3777	-11.7	-19.7	52	1.27	183	13	297.9	302.1	298.2
628.0	3837	-12.1	-29.1	23	0.55	181	14	298.2	300.1	298.3
600.0	4184	-14.9	-23.9	46	0.93	171	16	298.8	301.9	299.0
598.0	4210	-15.0	-22.7	52	1.04	170	16	299.0	302.4	299.2
589.0	4324	-15.5	-17.2	87	1.69	172	16	299.7	305.2	300.0

566.0	4624	-17.3	-21.0	73	1.27	178	15	301.0 305.2 301.3	
542.0	4946	-20.1	-22.3	82	1.18	185	15	301.5 305.4 301.7	
520.0	5253	-22.7	-23.6	92	1.10	175	16	301.9 305.6 302.1	
500.0	5540	-24.1	-26.4	81	0.89	165	17	303.6 306.6 303.8	
497.0	5583	-24.4	-26.9	80	0.86	165	18	303.7 306.6 303.9	
438.0	6494	-31.5	-36.5	61	0.39	174	19	305.9 307.3 306.0	
400.0	7130	-37.1	-42.1	60	0.24	180	20	306.7 307.6 306.7	
387.0	7352	-39.1	-44.0	59	0.20	180	23	307.0 307.7 307.0	
332.0	8383	-48.3	-52.9	59	0.09	156	32	308.1 308.4 308.1	
329.0	8442	-48.8	-53.5	58	0.08	155	33	308.2 308.5 308.2	
311.0	8808	-51.9	-56.9	55	0.06	155	29	308.9 309.1 308.9	
300.0	9040	-53.3	-59.3	48	0.04	155	27	310.1 310.3 310.1	
277.0	9552	-57.3	-64.3	41	0.02	162	28	311.5 311.6 311.5	
251.0	10175	-59.9	-66.9	40	0.02	170	29	316.5 316.6 316.5	
250.0	10200	-59.9	-66.9	40	0.02	170	29	316.9 317.0 316.9	
234.0	10613	-57.5	-65.5	35	0.02	162	21	326.6 326.7 326.6	
221.0	10968	-58.7	-68.1	29	0.02	155	15	330.1 330.2 330.2	
200.0	11590	-60.7	-72.7	19	0.01	155	17	336.5 336.5 336.5	
198.0	11653	-60.8	-72.8	19	0.01	155	17	337.2 337.3 337.2	
197.0	11684	-60.9	-72.9	19	0.01	157	17	337.6 337.7 337.6	
184.0	12112	-60.8	-74.0	16	0.01	180	12	344.5 344.6 344.5	
165.0	12794	-60.5	-75.8	12	0.01	270	5	355.8 355.9 355.8	
162.0	12908	-60.5	-76.1	11	0.01	270	5	357.8 357.8 357.8	
150.0	13390	-60.3	-77.3	9	0.01	180	10	366.0 366.0 366.0	
149.0	13432	-60.3	-77.3	9	0.01	175	11	366.7 366.7 366.7	
141.0	13776	-60.3	-77.3	9	0.01	204	14	372.5 372.6 372.5	
124.0	14572	-63.3	-82.0	6	0.00	270	20	381.0 381.0 381.0	
121.0	14724	-63.9	-82.9	6	0.00	272	20	382.6 382.6 382.6	
100.0	15900	-61.9	-81.9	5	0.00	285	22	407.9 407.9 407.9	
82.3	17105	-61.5	-82.5	5	0.01	289	24	432.0 432.1 432.0	
79.0	17356	-62.3	-83.5	4	0.00	290	24	435.5 435.6 435.5	
72.0	17927	-64.0	-85.8	4	0.00	265	28	443.6 443.6 443.6	
70.0	18100	-64.5	-86.5	3	0.00	270	29	446.1 446.1 446.1	
65.0	18549	-65.1	-87.3	3	0.00	295	41	454.3 454.3 454.3	
50.0	20140	-67.3	-90.3	3	0.00	275	48	484.5 484.5 484.5	
48.1	20373	-68.5	-92.5	2	0.00	275	47	487.0 487.0 487.0	
47.3	20473	-68.1	-92.1	2	0.00	275	47	490.3 490.3 490.3	

Station information and sounding indices

Station number: 2527

Observation time: 021204/0000

Station latitude: 57.66

Station longitude: 12.50

Station elevation: 164.0

Showalter index: 13.40

Lifted index: 17.44

LIFT computed using virtual temperature: 17.55

SWEAT index: 45.07
K index: -12.30
Cross totals index: 15.00
Vertical totals index: 19.80
Totals totals index: 34.80
Convective Available Potential Energy: 0.00
CAPE using virtual temperature: 0.00
Convective Inhibition: 0.00
CINS using virtual temperature: 0.00
Bulk Richardson Number: 0.00
Bulk Richardson Number using CAPV: 0.00
Temp [K] of the Lifted Condensation Level: 270.41
Pres [hPa] of the Lifted Condensation Level: 956.06
Mean mixed layer potential temperature: 273.92
Mean mixed layer mixing ratio: 3.29
1000 hPa to 500 hPa thickness: 5341.00
Precipitable water [mm] for entire sounding: 9.94

Table 8 Goteborg, Sweden Atmospheric Observations at 00Z 04 Dec 2002

PRES	HGHT	TEMP	DWPT	RELH	MIXR	DRCT	SKNT	THTA	THTE	THTV
hPa	m	C	C	%	g/kg	deg	knot	K	K	K
1003.0	164	-0.5	-1.6	92	3.41	150	5	272.4	281.8	273.0
1001.0	179	-0.9	-2.7	88	3.15	135	10	272.2	280.9	272.7
1000.0	186	-1.1	-3.2	86	3.03	130	11	272.1	280.4	272.6
977.0	370	-2.4	-3.9	89	2.94	120	20	272.6	280.8	273.1
935.0	717	-4.7	-5.3	96	2.77	132	20	273.6	281.4	274.1
925.0	802	-4.3	-5.0	95	2.86	135	20	274.9	282.9	275.4
918.0	862	-0.9	-5.1	73	2.86	135	21	279.0	287.2	279.5
914.0	897	-0.7	-5.7	69	2.75	135	21	279.5	287.4	280.0
902.0	1002	-1.2	-5.9	70	2.74	135	22	280.1	288.0	280.5
858.0	1400	-3.1	-6.7	77	2.72	150	18	282.1	290.0	282.6
850.0	1475	-3.5	-6.8	78	2.71	150	18	282.5	290.4	282.9
813.0	1825	-6.3	-7.8	89	2.63	147	18	283.1	290.8	283.6
792.0	2030	-5.3	-5.9	96	3.12	145	18	286.3	295.5	286.9
786.0	2090	-5.5	-6.1	95	3.10	145	18	286.8	295.9	287.3
715.0	2830	-7.6	-8.7	91	2.77	190	26	292.3	300.7	292.8
711.0	2874	-7.7	-8.9	91	2.76	187	26	292.6	301.0	293.1
700.0	2995	-8.9	-11.4	82	2.30	180	24	292.6	299.6	293.0
687.0	3140	-9.4	-12.6	78	2.12	165	21	293.6	300.1	293.9
681.0	3208	-9.7	-13.2	76	2.04	166	21	294.0	300.4	294.4
643.0	3646	-12.7	-14.9	84	1.88	170	21	295.4	301.4	295.8
617.0	3961	-14.9	-16.1	91	1.77	186	20	296.5	302.1	296.8

598.0	4197	-15.5	-19.7	70	1.35	199	19	298.4	302.8	298.7
596.0	4222	-15.7	-19.9	70	1.33	200	19	298.5	302.8	298.7
574.0	4500	-17.7	-22.4	67	1.12	195	17	299.3	303.0	299.5
500.0	5520	-25.3	-31.3	57	0.56	200	21	302.1	304.1	302.2
468.0	5996	-29.5	-37.5	46	0.33	203	25	302.7	303.9	302.7
431.0	6580	-33.3	-46.3	26	0.14	207	29	305.1	305.6	305.1
407.0	6979	-36.5	-48.0	30	0.12	210	32	305.9	306.4	305.9
400.0	7100	-37.5	-48.5	31	0.12	210	33	306.2	306.6	306.2
300.0	9020	-53.5	-61.5	37	0.03	195	43	309.8	310.0	309.8
292.0	9192	-55.1	-63.6	34	0.03	190	43	310.0	310.1	310.0
285.0	9347	-56.5	-65.5	31	0.02	194	44	310.1	310.2	310.1
268.0	9733	-58.3	-67.3	31	0.02	205	47	313.0	313.1	313.0
250.0	10170	-60.3	-69.3	30	0.01	205	44	316.3	316.4	316.3
238.0	10476	-61.5	-70.5	29	0.01	205	40	319.0	319.0	319.0
230.0	10689	-61.3	-71.0	26	0.01	205	36	322.4	322.5	322.4
209.0	11286	-60.6	-72.6	19	0.01	230	23	332.4	332.5	332.4
200.0	11560	-60.3	-73.3	17	0.01	230	19	337.1	337.2	337.1
188.0	11946	-58.1	-72.0	15	0.01	205	20	346.7	346.8	346.7
187.0	11980	-57.9	-71.9	15	0.01	207	20	347.5	347.6	347.5
172.0	12503	-58.4	-72.7	14	0.01	230	13	355.2	355.2	355.2
160.0	12956	-58.8	-73.5	13	0.01	225	8	361.9	362.0	361.9
150.0	13360	-59.1	-74.1	13	0.01	210	11	368.1	368.1	368.1
116.0	14964	-60.6	-75.0	13	0.01	295	12	393.3	393.4	393.3
112.0	15183	-60.8	-75.1	14	0.01	295	13	396.9	396.9	396.9
104.0	15645	-61.3	-75.4	14	0.01	260	11	404.5	404.6	404.5
100.0	15890	-61.5	-75.5	14	0.01	275	11	408.6	408.7	408.6
90.0	16538	-62.1	-76.2	14	0.01	285	20	419.8	419.9	419.8
77.0	17497	-63.1	-77.1	13	0.01	265	19	437.0	437.1	437.0
74.7	17683	-63.3	-77.3	13	0.01	268	23	440.4	440.5	440.4
70.0	18080	-64.7	-77.7	15	0.01	275	31	445.6	445.7	445.6
64.0	18623	-67.1	-80.1	14	0.01	290	30	451.9	452.0	451.9
60.3	18984	-68.7	-81.7	14	0.01	285	29	456.1	456.2	456.1
59.5	19065	-67.9	-80.9	14	0.01	282	31	459.6	459.7	459.6
55.0	19537	-68.4	-81.0	15	0.01	265	44	468.9	468.9	468.9
50.0	20110	-69.1	-81.1	16	0.01	265	55	480.2	480.3	480.2
45.1	20726	-68.9	-80.9	16	0.01	271	57	495.1	495.2	495.1
42.0	21148	-70.4	-82.4	16	0.01	275	59	501.6	501.7	501.6
36.5	21978	-73.3	-85.3	15	0.01	267	64	514.6	514.7	514.6
35.3	22174	-72.7	-84.7	15	0.01	265	65	521.0	521.1	521.0
35.0	22224	-72.6	-84.6	15	0.01	265	65	522.7	522.7	522.7
34.0	22394	-72.1	-84.1	15	0.01	260	61	528.3	528.4	528.3

Station information and sounding indices

Station number: 2527

Observation time: 021204/1200

Station latitude: 57.66

Station longitude: 12.50

Station elevation: 164.0
Showalter index: 10.63
Lifted index: 17.92
LIFT computed using virtual temperature: 17.98
SWEAT index: 57.11
K index: 12.50
Cross totals index: 18.50
Vertical totals index: 21.80
Totals totals index: 40.30
Convective Available Potential Energy: 0.00
CAPE using virtual temperature: 0.00
Convective Inhibition: 0.00
CINS using virtual temperature: 0.00
Bulk Richardson Number: 0.00
Bulk Richardson Number using CAPV: 0.00
Temp [K] of the Lifted Condensation Level: 268.79
Pres [hPa] of the Lifted Condensation Level: 950.45
Mean mixed layer potential temperature: 272.74
Mean mixed layer mixing ratio: 2.93
1000 hPa to 500 hPa thickness: 5334.00
Precipitable water [mm] for entire sounding: 12.03
Description of the indices.

Table 9 Goteborg, Sweden Atmospheric Observations at 12Z 04 Dec 2002

11.1 Smith-Weintraub Model Data Used in this Report

The Smith-Weintraub refractivity used to generate the AF results shown in Figure XXXX was calculated using average profiles for the meteorological terms in it as a function of altitude for mid-latitudes[21] (USA) for the Troposphere. It was necessary to create models of the temperature and various pressures as functions of altitude in order to calculate the Smith-Weintraub refractivity. The temperature profile was obtained from the Tiros Weather Satellite Operational Vertical Sounder results for the North Atlantic from the http://astro.pas.rochester.edu website of R. Benson's thesis *The Compression of a Monochromatic Database, and CPU Minimization Algorithms.*University of Rochester, 1996. Compare this profile with an actual one from Midland, texas from the University of Wyoming's meteorological database, Figure XXX.

Note how the Temperature and P wet profiles flatten out in the Troposphere between 10 and 20 km altitude, this is called the tropopause.

---

[21] American Geophysical Union, "Water vapor in the Atmosphere", Washington, DC, Special Report, December, 1995, Figure 1. Also, Sica, R.J., "Vertical Structure of the Earth's Temperature." University of Western Ontario, 1999.
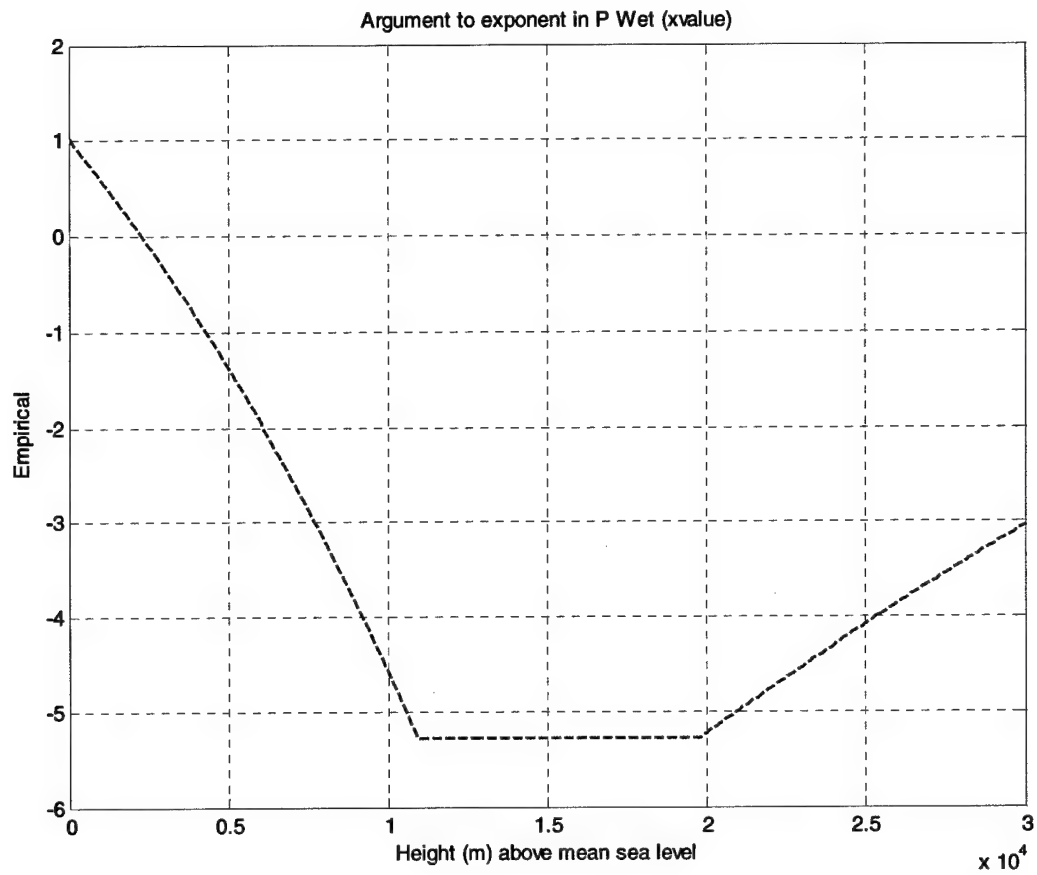
Modeled SW Argument to exponent in P wet
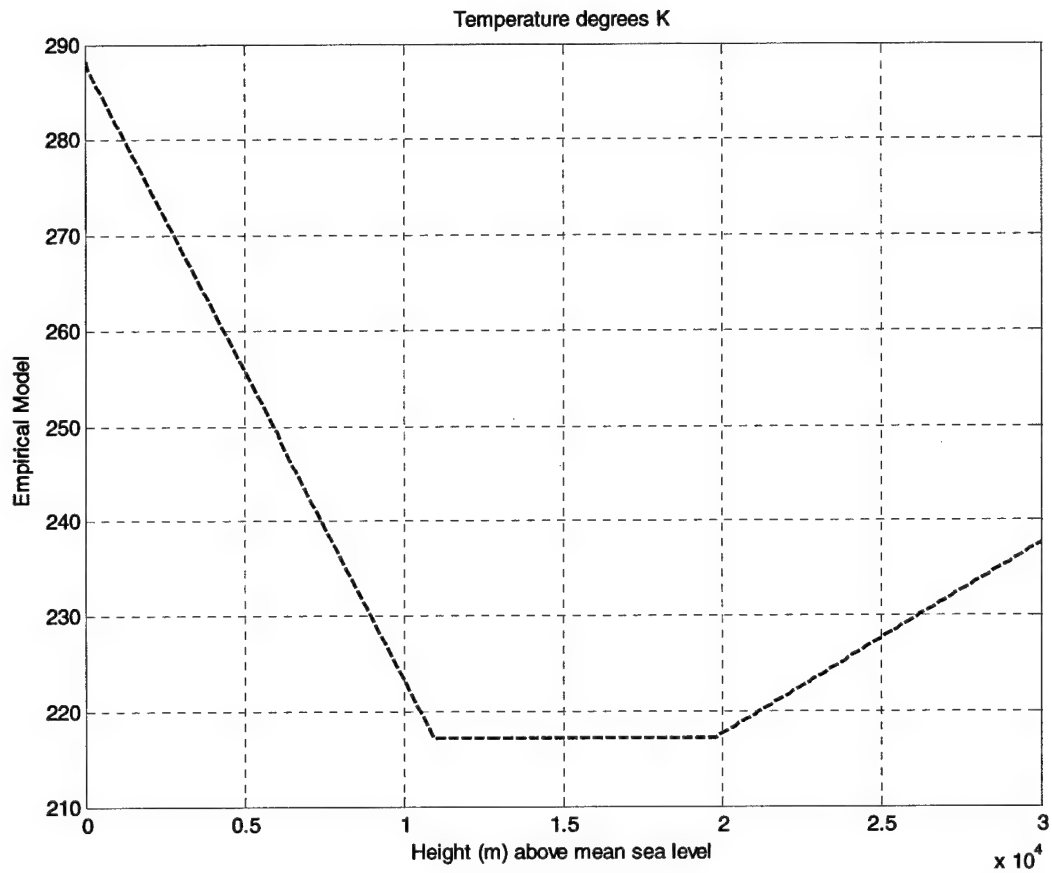
Modeled Temperature Profile

Temperature (degrees K) at 12Z04Dec2002
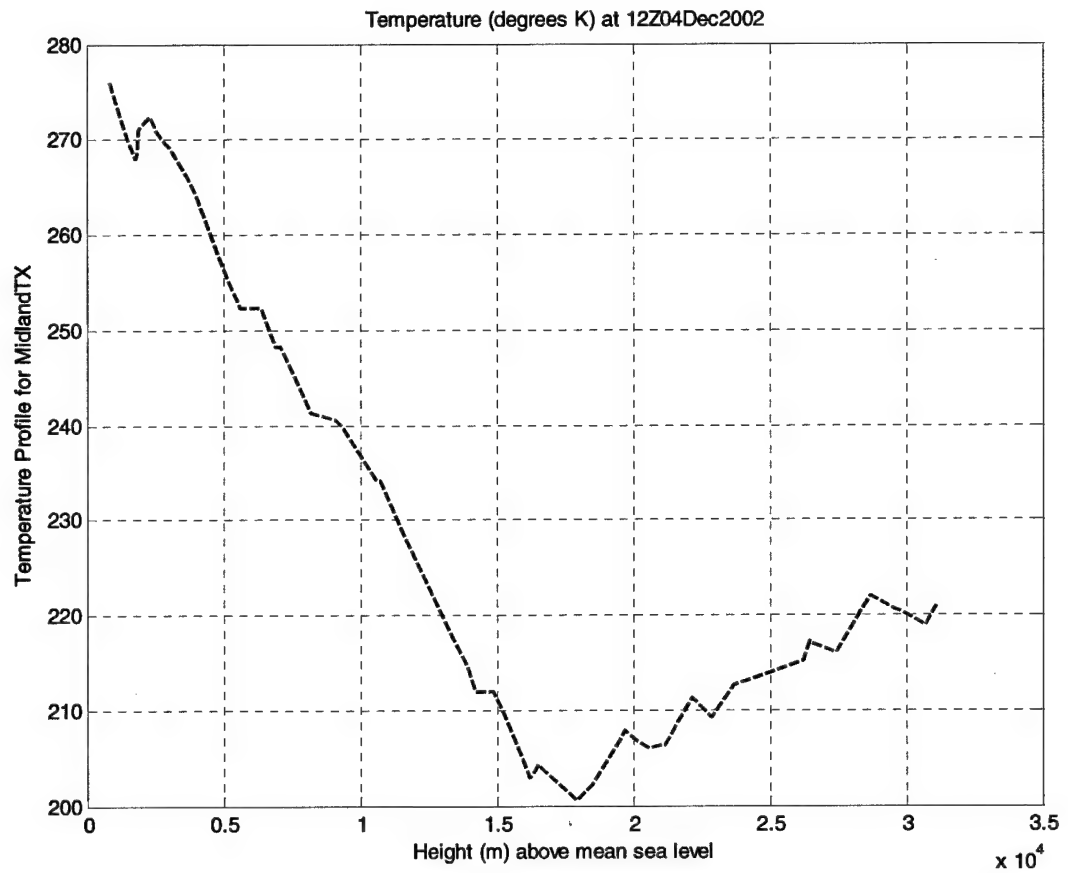
Actual Temperature Profile from Midland, Texas
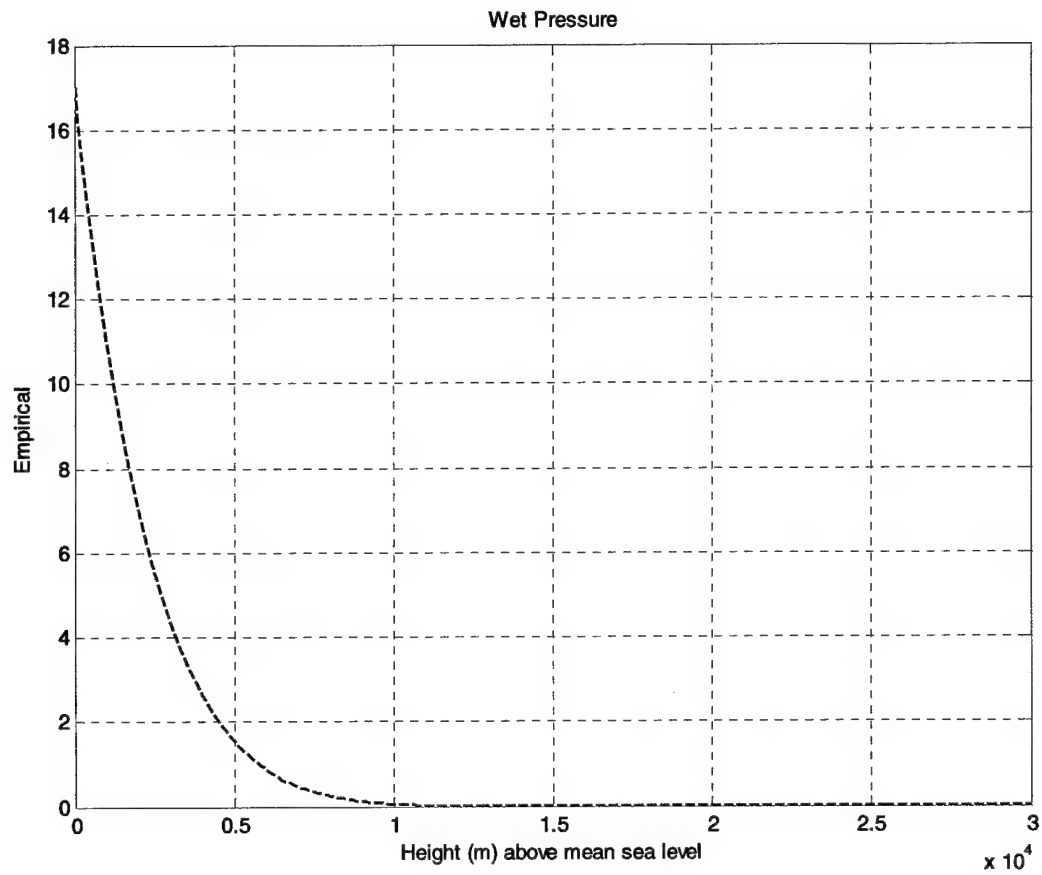
Modeled Wet Pressure (mbar)

Partial Wet Pressure (mbar) at 12Z04Dec2002
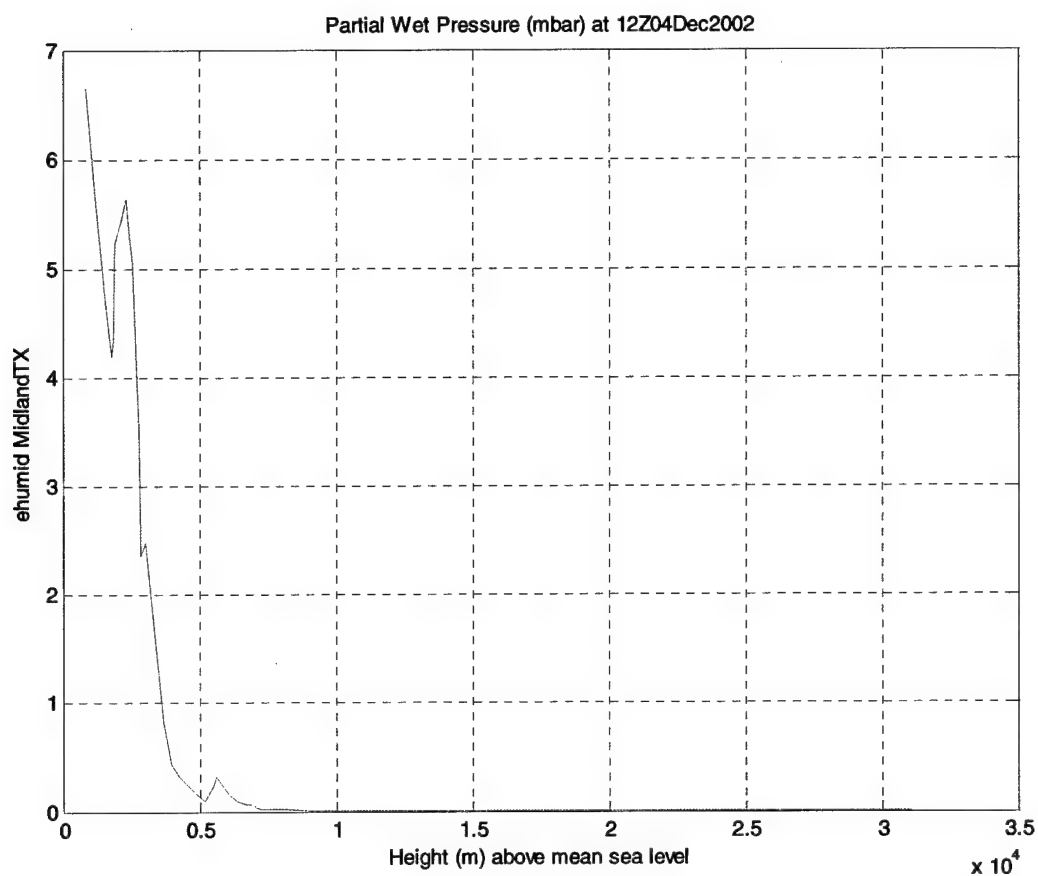
Actual Wet Pressure from Midland, Texas
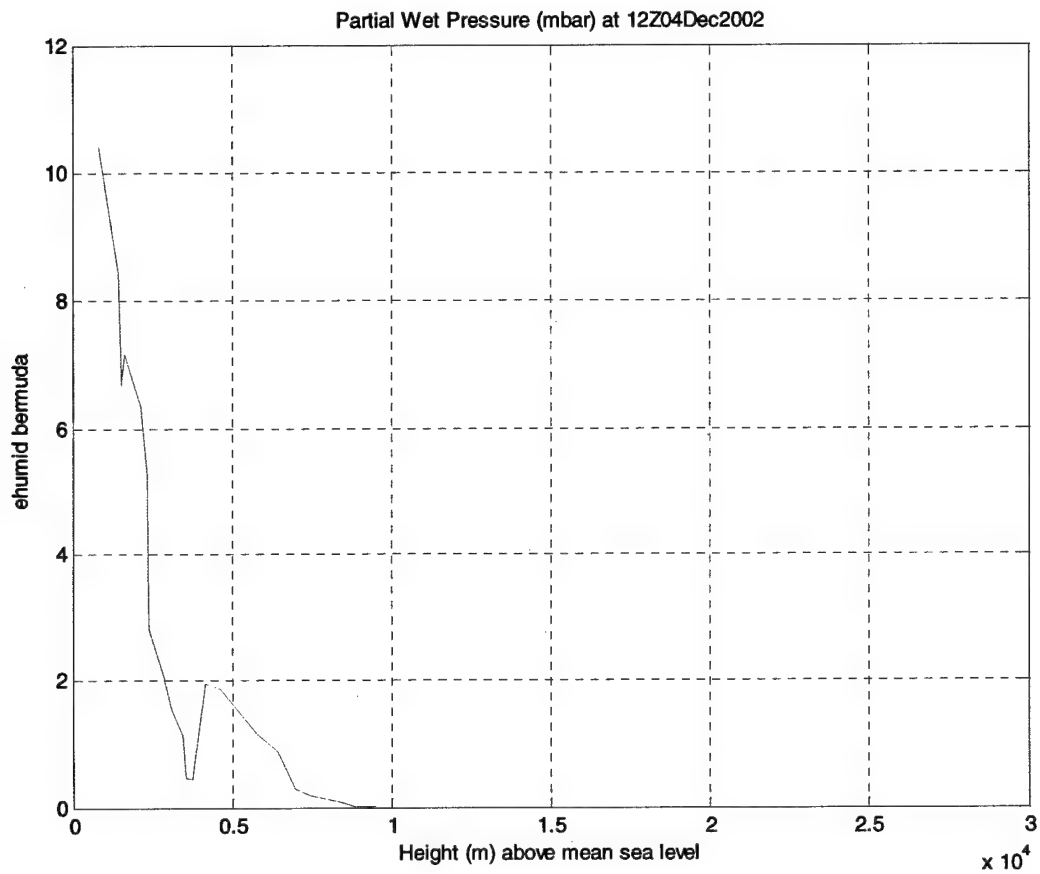
Actual Wet Pressure from Bermuda

Partial Wet Pressure (mbar) at 12Z04Jul2002
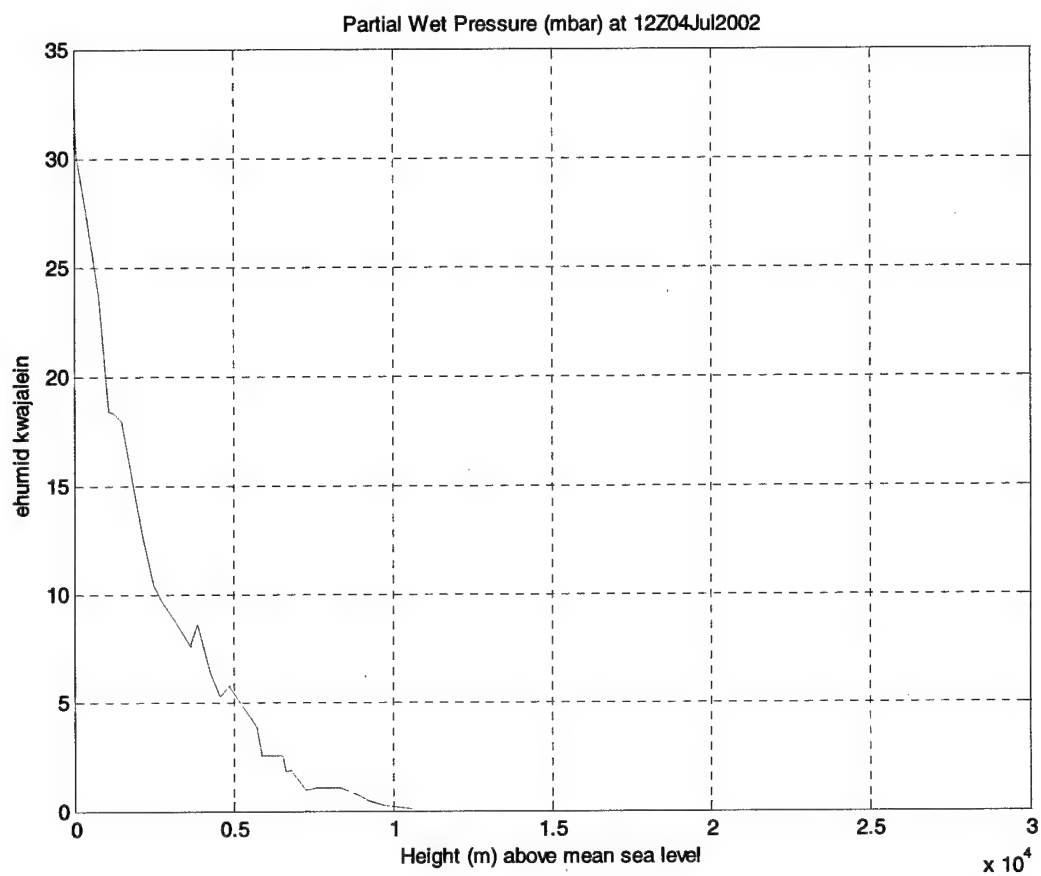
Actual Wet Pressure from Kwajalein Island, Pacific Ocean
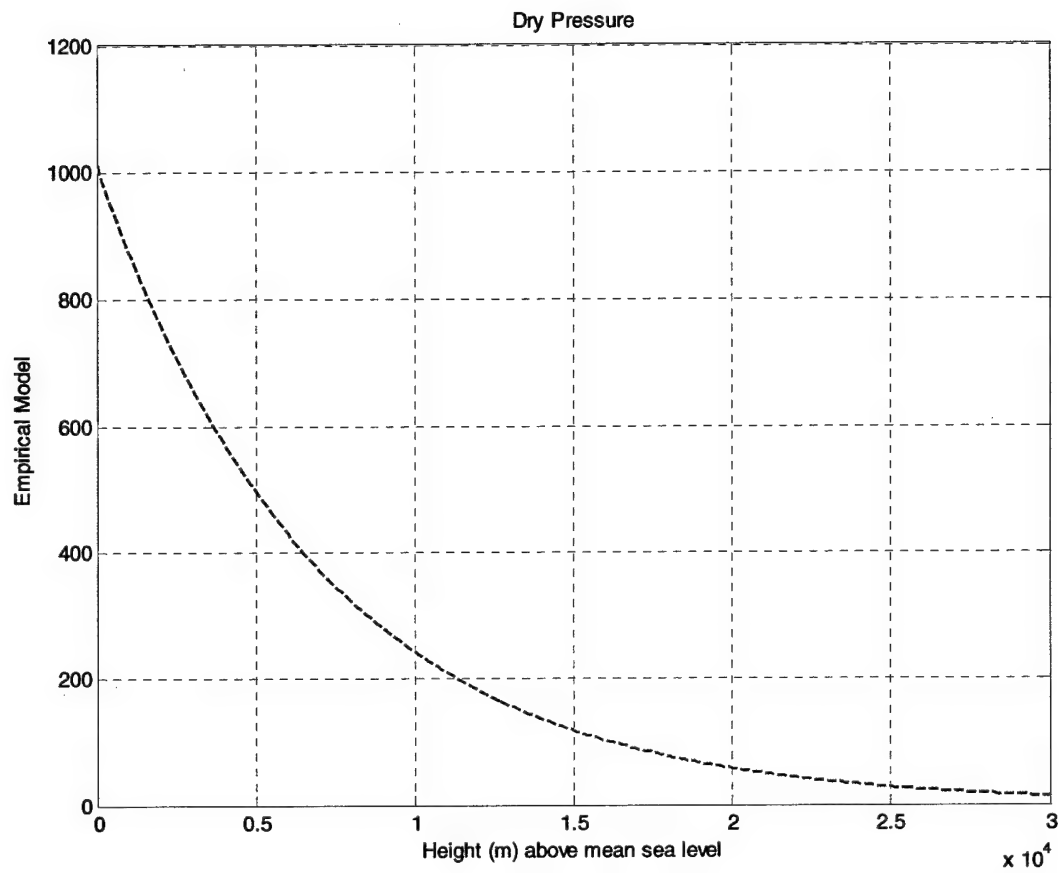
Modeled Dry Pressure

12 APPENDIX C, MATLAB Refractivity Routine

The following is the MATLAB routine used to generate the refractivity plots shown in Figures XXX to XXX. The input data are in files similar to Table 8. First, a linear fit is made to the semilog plot of the Smith-Weintraub refractivity. The y-intercept and slope of this are then used as the values of the parameters a and b in the exponential model of the refractivity, n. The parameter a is adjusted to shift the resulting exponential refractivity curve up or down to best effect a fit to the data at about 5 km altitude or thereabouts and higher, as we are generally more concerned with those levels rather than those closer to the ground. These parameters are then to be taken as examples of the numbers to be encountered in the exponential model.

```
% Data plotting routine for upper air data from file 'k'.data
  function plotprofile(k,c)
  c
h = [k,'.data'];
f = fullfile('C:','BAE','WyoData',c,h);
S =load (f,'-ascii');

figure;
plot(S(:,2),S(:,3)+273.16,'--b','LineWidth',2);
grid;
xlabel('Height (m) above mean sea level');
ylabel(['Temperature Profile for ',k]);
title(['Temperature (degrees K) at ',c]);


figure;
plot(S(:,2),S(:,1),'-.b','LineWidth',2);
grid;
xlabel('Height (m) above mean sea level');
ylabel(['Pressure Profile for ',k]);
title(['Pressure (hPa) at ',c]);


figure;
plot(S(:,2),S(:,5),'-.b','LineWidth',2);
grid;
xlabel('Height (m) above mean sea level');
ylabel(['Relative Humidity for ',k]);
title(['Relative Humidity (%) at ',c]);

        temp1 = 273.15;
    temp = S(:,3)+273.15;  % convert Celsius to Kelvin
    press = S(:,1); % total atmospheric pressure (mbar)
    ehumid = S(:,5); % relative humidity (%)
```

```
            xvalue1 = 25.22 * ((temp - temp1)./temp) - 5.31 * log( temp ./ temp1);
            ehumid1 = 6.105 * exp(xvalue1); % theoretical value
        %xvalue = (17.27 * S(:,3))./(S(:,3) + 237.3);
            %es = 6.108 * exp(xvalue); % saturation vapor pressure
        xvalue = (17.502 * S(:,3))./(temp - 32.18);
            es = 6.1121 * exp(xvalue); % saturation vapor pressure

        ehumid = es .* ehumid/100.  % humid air pressure in mbar
        temp2 = temp .* temp;
        temp3 = temp2 .* temp;
        tempc2 = S(:,3) .* S(:,3);
        tempc3 = tempc2 .* S(:,3);

            nswein = 77.6 * (press ./temp) + 3.73 * 100000.0 * (ehumid ./temp2);

K1 = 77.604 ;
K2 = 64.8 ;
K3 = 377600.0 ;
Zwinv = 1. + 1650. .*(ehumid ./temp3) .* (1. - 0.01317 .*  S(:,3) + 1.75*10^(-4) .* tempc2 + 1.44
.* 10^(-6) .* tempc3);
Zdinv = 1.0;
nthayer = K1 *((press –ehumid) ./temp)*Zdinv  + ( K2 *(ehumid ./temp) + K3 *(ehumid ./temp2 ))
.* Zwinv;


logp = polyfit(S(:,2),log(nswein),1)
logp(1)
logp(2)
%logp(3)* 10^-4
%%cheight = S(:,2);
%%sheight = (cheight - mean(cheight))./std(cheight);
logpred2 = exp(polyval(logp,S(:,2)) );

i = 0:100:30000;
%y1 = 10^6 * (0.000500*exp(-0.0001410999*i));
%y1 = 10^6 * ((logp(2)* 10^-4)*exp(logp(1)*i));
y1 = 10^6 * (0.000340*exp(logp(1)*i));
%y2 = 10^6 * (0.000800*exp(-0.000320*i));
figure;
%h = plot(i,y1,S(:,2),nswein,S(:,2),nthayer,i,y2);
%h = semilogy(S(:,2),logpred2,S(:,2),nswein,S(:,2),nthayer);
%h = plot(S(:,2),logpred2,S(:,2),nswein,S(:,2),nthayer);
h = plot(i,y1,S(:,2),nswein,S(:,2),nthayer);
set(h(1),'LineStyle','-','Color','b');
set(h(2),'LineStyle','-.','Color','r');
set(h(3),'LineStyle',':','Color','m');
%set(h(4),'LineStyle','-','Color','g');
```

```
grid;
xlabel('Height (m) above mean sea level');
ylabel(['Refractivity Profile for ',k]);
title(['Refractivity at ',c]);
[legend_h,object_h,plot_h,text_strings] = legend(' Exponential Model',' Smith Weintraub Model','
Thayer Model');

%NOTES:
%The Thayer Model, N = K1 (Pd / T) Zd-1 + [(K2 (e/T) + K3 (e/T2)] * ZW-1.  Where Pd is
%the partial pressure of the dry gases in the atmosphere, e is the partial pressure
%of the water vapor (ehumid?), T is the absolute temperature, Zd is the compressibility factor for
dry
% air, Zw is the compressibility factor for water vapor, and Ki are constants empirically
%determined.
% Zd-1 = 1.0
% From ref. Rocken, et al, "Improved Mapping of Tropospheric Delays", 2000:
% ZW-1 = 1. + 1650. *(e/T^3) * (1. - 0.01317* t + 1.75*10^(-4) * t^2 + 1.44 * 10^(-6) * t^3)
% where t is temperature in degrees Celsius. T in Kelvin.
% K1 = 77.604 degrees K/mbar
% K2 = 64.8 degrees K/mbar
% K3 = 377600 (degrees K)^2/mbar

%figure;
%h = plot(S(:,2),ehumid,S(:,2),ehumid1);
%set(h(1),'LineStyle','-','Color','b');
%set(h(2),'LineStyle','-.','Color','r');
%grid;
%xlabel('Height (m) above mean sea level');
%ylabel(['Relative Humidity for ',k]);
%title(['Relative Humidity (%) at ',c]);
%[legend_h,object_h,plot_h,text_strings] = legend(' Model1',' Model2');
```

Refractivity at 12Z04Dec2002

Exponential and Smith-Weintraub Models

Wet Pressure
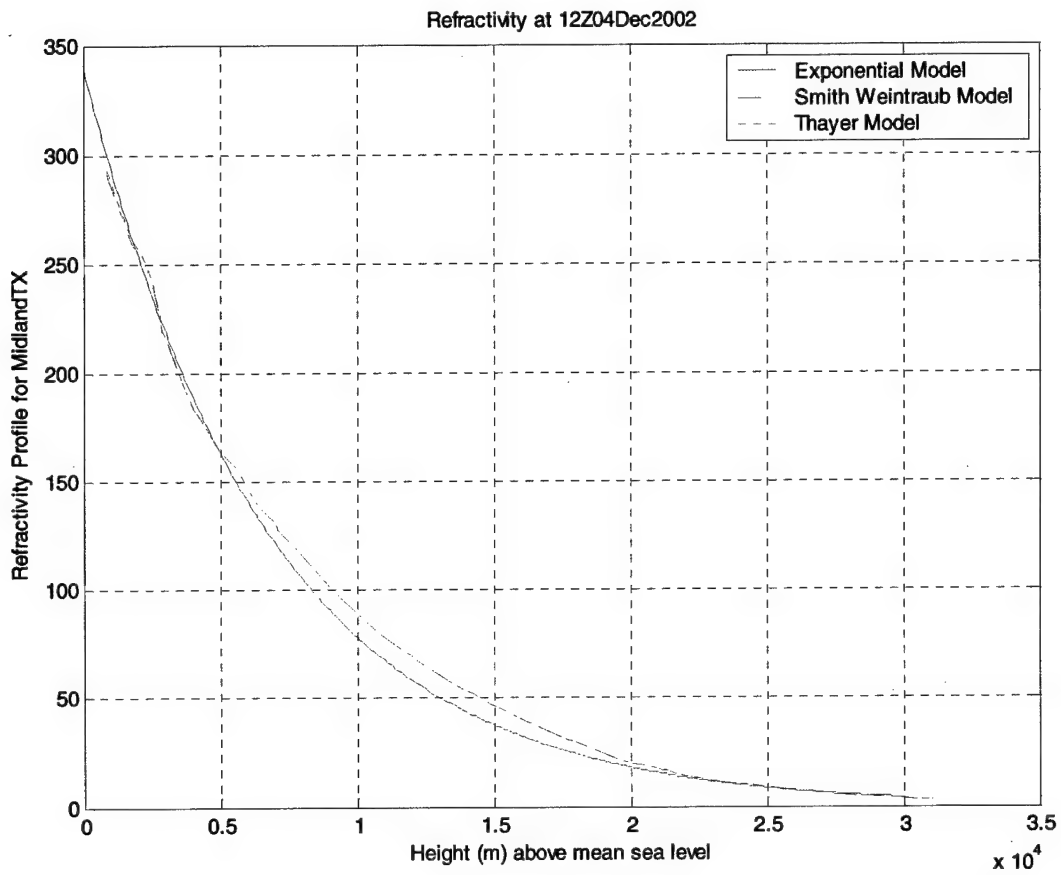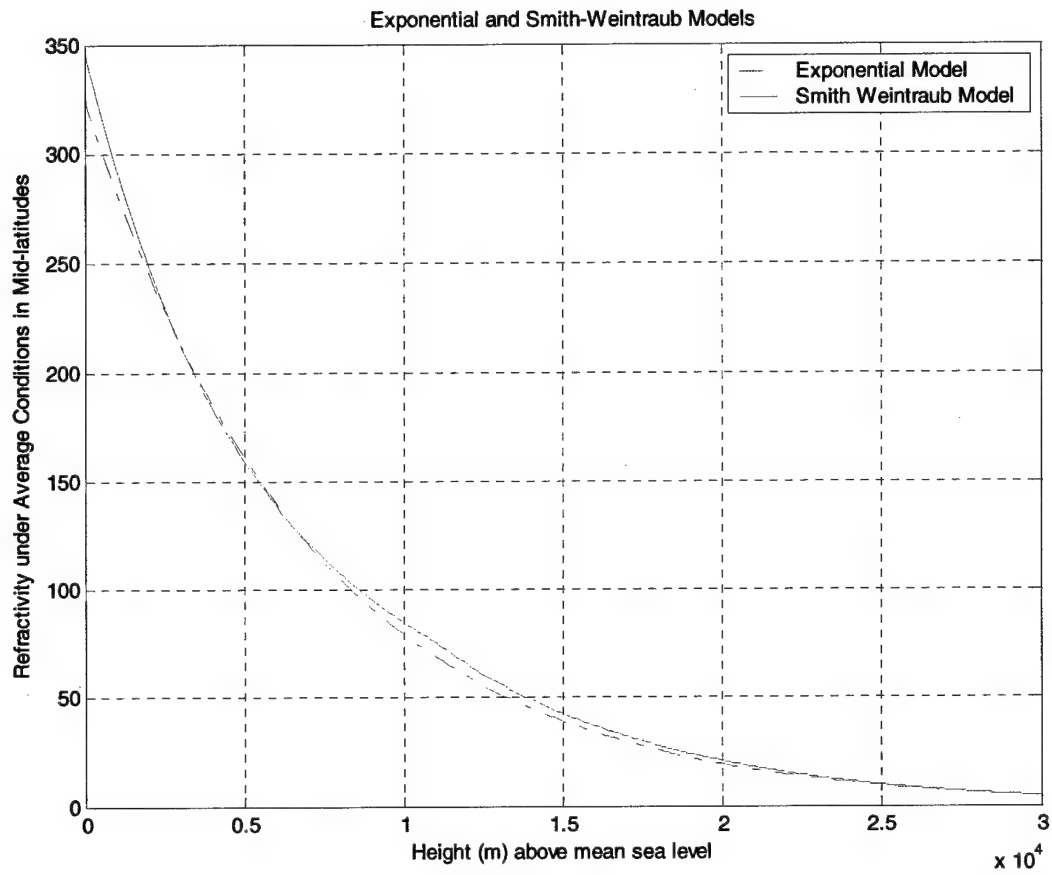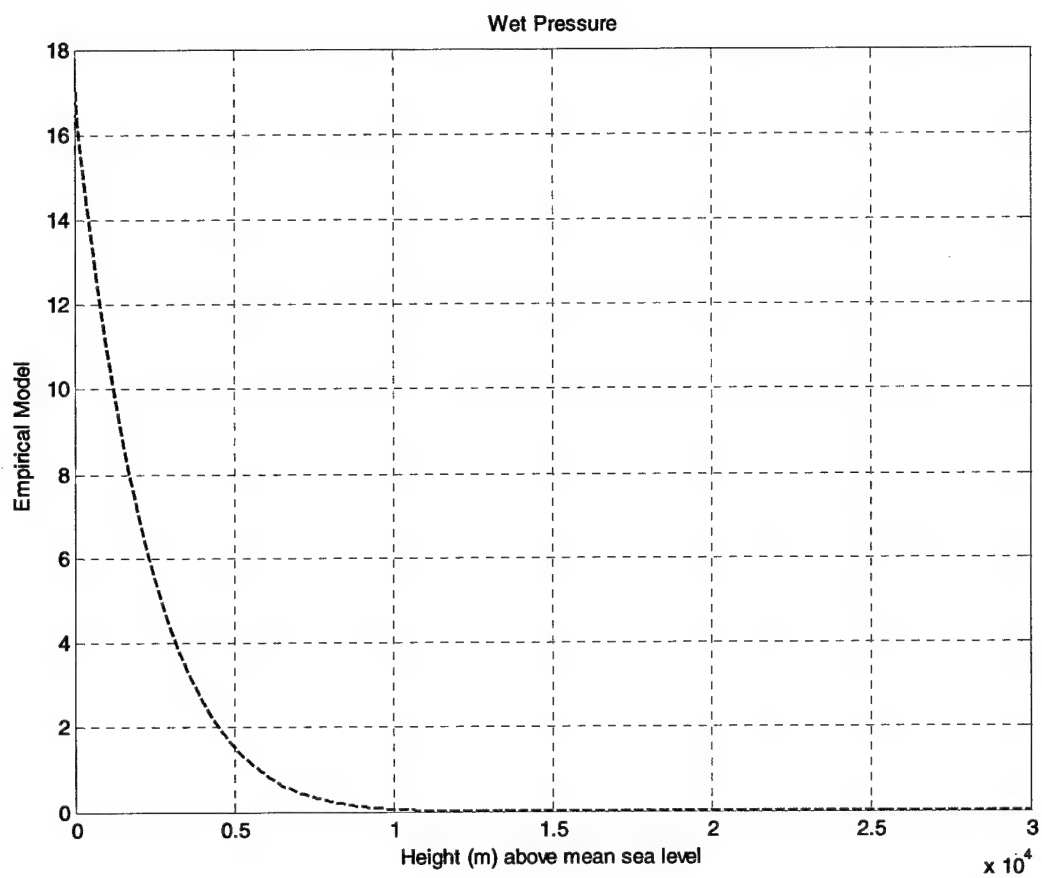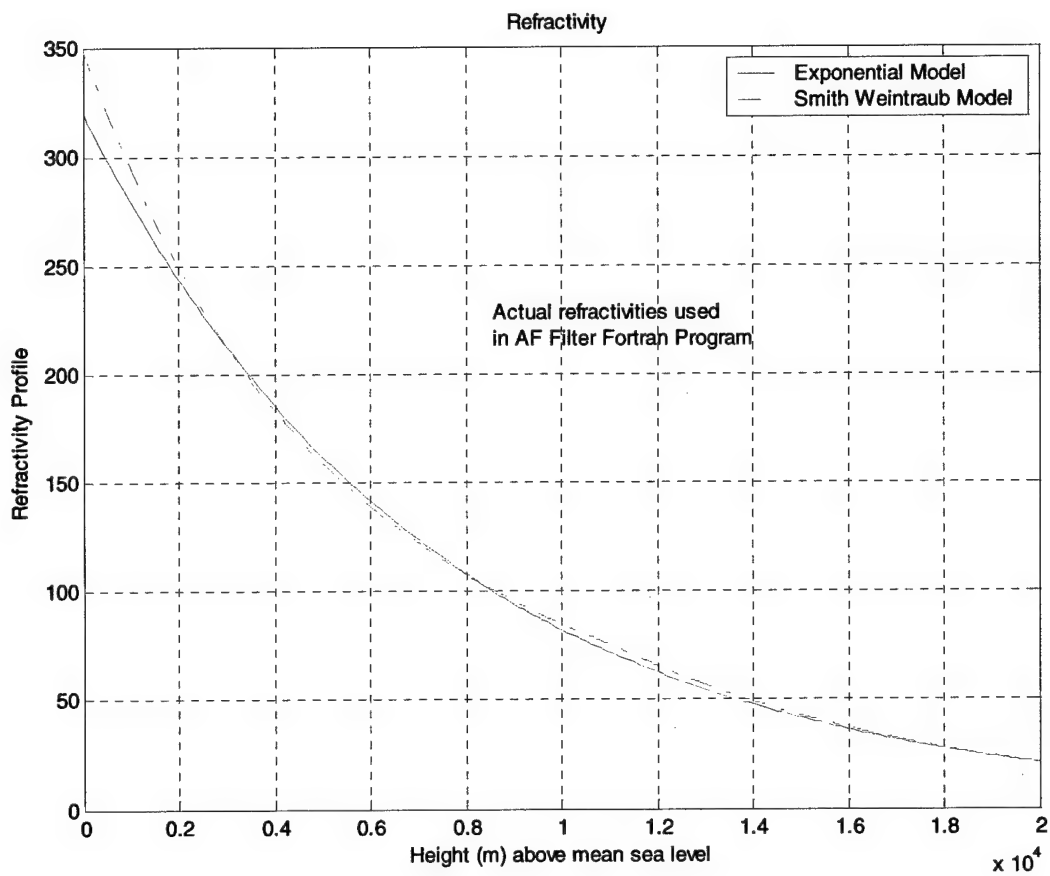
Figure 82AF Test Program Refractivities

13 APPENDIX D Atmospheric Filter Kalman Filter Fortran Program

```
C-------------------------------------------------------------------
C         ATMOSPHERIC CALIBRATION FILTER TESTBED
C
C     version of 6/03 to generate results for AF Final Report 7/03
C-------------------------------------------------------------------
      PROGRAM ATMOS
      USE IMSLF90   !added 9/25/02
      IMPLICIT NONE
C*********************************************************************
C
C         DEFINE TRUE WORLD VARIABLES AND CONSTANTS
C
C*********************************************************************

      REAL*8 ATRUE,BTRUE,PI
      REAL*8 GMTRU,ALFTRU
      REAL*8 LAT(4),LON(4),H(4)
      REAL*8 RE,CM,RAD,ELEV
      REAL*8 THETA0,PHI0,S0,SE,LE,LC,K,KC,TCOM
      REAL*8 TAUM,RSL,SIG,SIG2,DCDA,DCDB,DLDG,DLDA,FAC
      REAL*8 S0EST,S0ERR,SGS0FE,SGS0RE
      CHARACTER(40) timest
      Logical done,debug
C*********************************************************************
C
C         DEFINE VARIABLES FOR FLAT EARTH KALMAN FILTER ESTIMATION
C
C*********************************************************************

      REAL*8 PF(2,2),QF(2,2),XF(2),RF(2,2),PHI(2,2),HF(2,2)
      REAL*8 AF,BF
      REAL*8 YF(2),YFEXP(2)
      REAL*8 ERRAF,ERRBF
      REAL*8 TIME,DELT,SGA0,SGB0,SGA,SGB,A0,B0,TIMEF


C*********************************************************************
C
C         DEFINE VARIABLES FOR RAY ESTIMATION KALMAN FILTER ESTIMATION
C
C*********************************************************************

      REAL*8 PR(3,3),QR(3,3),XR(3),RR(3,3),HR(3,3)
      REAL*8 AR,BR,GAMMA,ALPHA,GAM0,ALF0
      REAL*8 YR(3),YREXP(3)
      REAL*8 ERRGAM,ERRALF,LEA,B,LCA
      REAL*8 LEA1,LEA2,LEA3,LEA4,LEA5   !for testing purposes
      REAL*8 ERRAR,ERRBR,RNG,RNGX,RNGY,RNGZ,RNG3
      REAL*8 SGGM0,SGALF0
      REAL*8 SGK0 ! initial variance for K in the new RE model 10/7/02


C*********************************************************************
C
C         DEFINE OTHER VARIABLES
C
C*********************************************************************

      INTEGER*4 I,J,MDL,ACNUM,NFILE,IMBR,S0FILE,S0FILE1,S0INDX
```

```fortran
       REAL*8 PARAMS(34,5),ERF2
       CHARACTER (10) TYPE(34)
      EXTERNAL DERF   !imsl routine
      REAL*8  DERF,LONINC,THETA03,PHI03,PHIEST
      REAL*8 K0 ! initial value

      REAL*8 LAT0(4),LON0(4),H0(4)
      REAL*8 TIME0,DELT0,TIMEF0,S0SAVED

      REAL*8 oldaval,oldab2,oldbval,L16err1,L16FACTR,L16err2

C ****************************************************

      DATA PI/3.141592653589793D0/
      DATA RE/6378137.0D0/
      DATA CM/2.99792458E8/
      DATA RAD/57.2957795131D0/
      DATA KC/3.747405725D0/
      DATA K0/1.00000D0/   ! initial value
      DATA LCA/0.00D0/

      DATA done/.FALSE./
      DATA debug/.TRUE./
      DATA S0FILE,S0FILE1/50,51/

c     SCENARIOS:

C ****************************************************

C ****************************************************
c     Range 1 to 3= 9 km, theta0 = 3 degrees SCENARIO:  Short/Low, in medium
altitude
c     DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,10000.0D0/
c     DATA LAT(2),LON(2),H(2)/45.0610D0,45.0800D0,10600.0D0/
c     DATA LAT(3),LON(3),H(3)/45.0600D0,45.0810D0,10500.0D0/
c     DATA LAT(4),LON(4),H(4)/45.0620D0,45.0800D0,10400.0D0/
c
c     DATA SGGM0,SGALF0/25500.0,20.0/
c     DATA SGK0/30.0/
c     DATA GAM0,ALF0  /2066.0,38.0/    !new values = /E(gamma),E(alpha)/
c     DATA K/1.330000D0/  ! initial value
c     DATA SGA0,SGB0/100.0E-6,1.0E-3/
c     DATA A0,B0/350.0E-6,0.15E-3/
C ****************************************************
c
c
C ****************************************************
c     Range 1 to 3= 7 km, theta0 = 3 degrees SCENARIO:  Short/Low  low altitude
c     DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,100.0D0/
c     DATA LAT(2),LON(2),H(2)/45.0600D0,45.0800D0,600.0D0/
c     DATA LAT(3),LON(3),H(3)/45.0500D0,45.0500D0,500.0D0/
c     DATA LAT(4),LON(4),H(4)/45.0600D0,45.0800D0,400.0D0/
c
c     DATA SGGM0,SGALF0/18000.0,4.4/
c     DATA SGK0/1400.0/
c     DATA GAM0,ALF0   /2066.0,38.0/
c     DATA K/1.330000D0/
```

```
c      DATA SGA0,SGB0/400.0E-6,1.0E-4/
c      DATA A0,B0/350.0E-6,0.15E-4/
C ********************************************************
c
c
C ********************************************************
c      Range 1 to 3= 9 km, theta0 = 32 degrees SCENARIO: Short/Medium test
c      DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,10000.0D0/
c      DATA LAT(2),LON(2),H(2)/45.0600D0,45.0800D0,10100.0D0/
c      DATA LAT(3),LON(3),H(3)/45.0500D0,45.0500D0,10000.0D0/
c      DATA LAT(4),LON(4),H(4)/45.0600D0,45.0800D0,10000.0D0/

C ********************************************************
c
c
c
C ********************************************************
c      Range 1 to 3= 5 km, theta0 = 45 degrees SCENARIO: Short/HIGH
c      DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,10000.0D0/
c      DATA LAT(2),LON(2),H(2)/45.00200D0,45.00200D0,15000.0D0/
c      DATA LAT(3),LON(3),H(3)/45.00200D0,45.00200D0,15000.0D0/
c      DATA LAT(4),LON(4),H(4)/45.00200D0,45.00200D0,16000.0D0/
c      DATA SGGM0,SGALF0/2940.0,3600.0/
c      DATA SGK0/2.0/
c      DATA GAM0,ALF0  /2551.00,10.0/    !sort of OK 6/26 ????
c      DATA K/1.330000D0/  ! initial value
c      DATA SGA0,SGB0/400.0E-4,1.0E-3/  ! calculated values
c      DATA A0,B0/350.0E-5,0.15E-3/
C ********************************************************
c
C ********************************************************
c      Range 1 to 3= 9 km, theta0 = 10 degrees SCENARIO:  SHORT/MED
       DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,24500.0D0/
       DATA LAT(2),LON(2),H(2)/45.0600D0,45.0800D0,25550.0D0/
       DATA LAT(3),LON(3),H(3)/45.0500D0,45.0530D0,26000.0D0/
       DATA LAT(4),LON(4),H(4)/45.0800D0,45.0800D0,24900.0D0/

       DATA SGGM0,SGALF0/2940.0,96.0/
       DATA SGK0/1.0/
       DATA GAM0,ALF0  /2551.00,38.0/    !new values = /E(gamma),E(alpha)/
       DATA K/1.0000D0/  ! initial value
       DATA SGA0,SGB0/400.0E-6,1.0E-3/  ! calculated values
       DATA A0,B0/350.0E-6,0.15E-3/
C ********************************************************
c
c
C ********************************************************
c      Range 1 to 3= 60 km, theta0 = 1.0 degrees SCENARIO:  MEDIUM/LOW
c      DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,23600.0D0/
c      DATA LAT(2),LON(2),H(2)/45.4600D0,45.4800D0,25000.0D0/
c      DATA LAT(3),LON(3),H(3)/45.4500D0,45.4500D0,25000.0D0/
c      DATA LAT(4),LON(4),H(4)/45.4800D0,45.4800D0,25000.0D0/
c
c      DATA SGGM0,SGALF0/2940.0,3600.0/
c      DATA SGK0/0.5/
c      DATA GAM0,ALF0  /2551.00,400.0/
c      DATA K/1.330000D0/  ! initial value
```

```
c       DATA SGA0,SGB0/400.0E-7,1.0E-5/
c       DATA A0,B0/350.0E-6,0.15E-3/
C ****************************************************
c
C ****************************************************
c       Range 1 to 3= 60 km, theta0 = -10 degrees SCENARIO:  MEDIUM/MEDIUM
c       DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,35000.0D0/
c       DATA LAT(2),LON(2),H(2)/45.450D0,45.4200D0,25500.0D0/
c       DATA LAT(3),LON(3),H(3)/45.450D0,45.4200D0,25500.0D0/
c       DATA LAT(4),LON(4),H(4)/45.450D0,45.4200D0,25500.0D0/
c
c       DATA SGGM0,SGALF0/2940.0,3600.0/
c       DATA SGK0/2000.0/
c       DATA GAM0,ALF0   /2551.00,450.0/    !new values = /E(gamma),E(alpha)/
c       DATA K/1.330000D0/   ! initial value
c       DATA SGA0,SGB0/400.0E-7,1.0E-5/   ! calculated values
c       DATA A0,B0/350.0E-6,0.15E-3/
C ****************************************************
c
c
C ****************************************************
c       Range 1 to 3= 22 km, theta0 = 45 degrees SCENARIO:Medium/High
c       DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,0000.0D0/
c       DATA LAT(2),LON(2),H(2)/45.0600D0,45.0800D0,25550.0D0/
c       DATA LAT(3),LON(3),H(3)/45.0500D0,45.0530D0,26000.0D0/
c       DATA LAT(4),LON(4),H(4)/45.0800D0,45.0800D0,24900.0D0/
c
c       DATA SGGM0,SGALF0/2940.0,3600.0/
c       DATA SGK0/2.0/
c       DATA GAM0,ALF0   /2551.00,400.0/    !sort of OK 6/26 ????
c
c       DATA K/1.330000D0/   ! initial value
c       DATA SGA0,SGB0/400.0E-7,1.0E-5/   ! calculated values
c       DATA A0,B0/350.0E-6,0.1306E-3/
C ****************************************************
c
c
C ****************************************************
c       Range 1 to 3= 145 km, theta0 = 0.50 degrees SCENARIO:Long/Low
c       DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,9000.0D0/
c       DATA LAT(2),LON(2),H(2)/46.0600D0,46.0800D0,12000.0D0/
c       DATA LAT(3),LON(3),H(3)/46.0500D0,46.0500D0,12000.0D0/
c       DATA LAT(4),LON(4),H(4)/46.0600D0,46.0800D0,12000.0D0/
c
c       DATA SGGM0,SGALF0/2940.0,3600.0/
c       DATA SGK0/2000.0/
c       DATA GAM0,ALF0   /2551.00,450.0/    !new values = /E(gamma),E(alpha)/
c       DATA K/1.330000D0/   ! initial value
c       DATA SGA0,SGB0/400.0E-7,1.0E-5/   ! calculated values
c       DATA A0,B0/350.0E-6,0.15E-3/
C ****************************************************
c
c
C ****************************************************
c       Range 1 to 3= 225 km, theta0 = 10 degrees SCENARIO: Long/Higher
c       DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,10000.0D0/
c       DATA LAT(2),LON(2),H(2)/46.6200D0,46.6310D0,47000.0D0/
```

```
c        DATA LAT(3),LON(3),H(3)/46.6000D0,46.6300D0,45100.0D0/
c        DATA LAT(4),LON(4),H(4)/46.5998D0,46.6320D0,44900.0D0/
c
c        DATA SGGM0,SGALF0/7940.0,8603.0/
c        DATA SGK0/2000.0/
c        DATA GAM0,ALF0  /2000.00,10.0/    !new values = /E(gamma),E(alpha)/
c        DATA K/1.330000D0/   ! initial value
c        DATA SGA0,SGB0/400.0E-7,1.0E-5/   ! calculated values
c        DATA A0,B0/350.0E-6,0.15E-3/
C ****************************************************
c
C ****************************************************
c    Range 1 to 3= 150 km, SCENARIO: Table 4 L16 compensation calculation
c        DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,20000.0D0/
c        DATA LAT(2),LON(2),H(2)/46.0600D0,46.0800D0,20000.0D0/
c        DATA LAT(3),LON(3),H(3)/46.0900D0,46.1000D0,20000.0D0/
c        DATA LAT(4),LON(4),H(4)/46.0600D0,46.0800D0,20000.0D0/
c
C ****************************************************
c    Range 1 to 3= 250 km, SCENARIO: Table 4 L16 compensation calculation
c        DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,20000.0D0/
c        DATA LAT(2),LON(2),H(2)/46.8200D0,46.9000D0,20000.0D0/
c        DATA LAT(3),LON(3),H(3)/46.8200D0,46.9000D0,20000.0D0/
c        DATA LAT(4),LON(4),H(4)/46.8200D0,46.9000D0,20000.0D0/
c
C ****************************************************
c    Range 1 to 3= 350 km, SCENARIO: Table 4 L16 compensation calculation
c        DATA LAT(1),LON(1),H(1)/45.0200D0,45.0200D0,20000.0D0/
c        DATA LAT(2),LON(2),H(2)/47.5100D0,47.8000D0,15000.0D0/
c        DATA LAT(3),LON(3),H(3)/47.5100D0,47.8000D0,15000.0D0/
c        DATA LAT(4),LON(4),H(4)/47.5100D0,47.8000D0,15000.0D0/
c
C ****************************************************
c    Range 1 to 3= 450 km, SCENARIO: Table 4 L16 compensation calculation
c        DATA LAT(1),LON(1),H(1)/45.0200D0,45.0200D0,10000.0D0/
c        DATA LAT(2),LON(2),H(2)/48.5000D0,48.0000D0,15000.0D0/
c        DATA LAT(3),LON(3),H(3)/48.5000D0,48.0000D0,15000.0D0/
c        DATA LAT(4),LON(4),H(4)/48.5000D0,48.0000D0,15000.0D0/
c
C ****************************************************
c    Range 1 to 3= 550 km, SCENARIO: Table 4 L16 compensation calculation
c        DATA LAT(1),LON(1),H(1)/45.2000D0,45.4300D0,20000.0D0/
c        DATA LAT(2),LON(2),H(2)/49.5000D0,49.0000D0,20000.0D0/
c        DATA LAT(3),LON(3),H(3)/49.5000D0,49.0000D0,20000.0D0/
c        DATA LAT(4),LON(4),H(4)/49.5000D0,49.0000D0,20000.0D0/
c
C ****************************************************
c    Range 1 to 3= 475 km, theta0 = 3 degrees SCENARIO: Very Long/Low
c
c        DATA LAT(1),LON(1),H(1)/45.0000D0,45.0000D0,25000.0D0/
c        DATA LAT(2),LON(2),H(2)/48.5000D0,48.5000D0,45000.0D0/
c        DATA LAT(3),LON(3),H(3)/48.5100D0,48.5100D0,45000.0D0/
c        DATA LAT(4),LON(4),H(4)/48.5000D0,48.5000D0,45000.0D0/
c
c        DATA K/1.000000D0/   ! initial value
c        DATA SGGM0,SGALF0/7940.0,8603.0/
c        DATA SGK0/2000.0/
```

```
c      DATA GAM0,ALF0  /2551.00,400.0/    !new values = /E(gamma),E(alpha)/
c      DATA SGA0,SGB0/461.8E-8,1.1306E-4/
c      DATA A0,B0/350.0E-6,0.15E-3/
C ****************************************************
C ****************************************************
c
c
C ********************************************************
c      Other Initializations
C ********************************************************
       DATA ATRUE/326.60E-6/     ! from Smith-Weintraub Model
       DATA BTRUE/0.13061E-3/    ! from Smith-Weintraub Model
       DATA LONINC/0.00D0/  !added 10/18/2002, LON increment in degrees
       DATA TIME,DELT,TIMEF/0.0,1.0,200.0/   ! seconds
       DATA MDL/1/
C          MDL = 1, EXPONENTIAL MODEL, 2,SMITH_WEINTRAUB MODEL
       DATA ACNUM/3/
C            ACNUM = show results for this a/c number

       DATA TYPE/'PHI0','S0','SE','LE','LC','THETA0',
      1      'TCOM','TAUM','HF11','HF12','YF1','YEXP1',
      2      'RSL','PF11','PF22','K','XR1','XR2',
      3      'TCOM(RE)','LEA','LCA','YR1','YREXP1','RSL',
      4      'PR12','PR21','HR11','HR12','TIME','S0EFE','S0ERE',
      5      'SGS0FE','SGS0RE','XF1'/

C********************************************************************
C       DATA FILE SPECIFICATIONS
C********************************************************************

       OPEN(UNIT = 9, FILE = 'M:\FORTRAN\DATA\AF.DATA')
       OPEN(UNIT = 10,FILE = 'M:\FORTRAN\DATA\BF.DATA')
       OPEN(UNIT = 11,FILE = 'M:\FORTRAN\DATA\SIGAF.DATA')
       OPEN(UNIT = 12,FILE = 'M:\FORTRAN\DATA\SIGBF.DATA')
       OPEN(UNIT = 13,FILE = 'M:\FORTRAN\DATA\GAM.DATA')
       OPEN(UNIT = 14,FILE = 'M:\FORTRAN\DATA\ALF.DATA')
       OPEN(UNIT = 15,FILE = 'M:\FORTRAN\DATA\SGS0FE.DATA')
       OPEN(UNIT = 16,FILE = 'M:\FORTRAN\DATA\SGS0RE.DATA')
       OPEN(UNIT = 17,FILE = 'M:\FORTRAN\DATA\RSLFE.DATA')
       OPEN(UNIT = 18,FILE = 'M:\FORTRAN\DATA\RSLRE.DATA')
       OPEN(UNIT = 19,FILE = 'M:\FORTRAN\DATA\SIGGAM.DATA')
       OPEN(UNIT = 20,FILE = 'M:\FORTRAN\DATA\SIGALF.DATA')
       OPEN(UNIT = 21,FILE = 'M:\FORTRAN\DATA\S0EFE.DATA')
       OPEN(UNIT = 22,FILE = 'M:\FORTRAN\DATA\S0ERE.DATA')
C      OPEN(UNIT = 23,FILE = 'M:\FORTRAN\DATA\N.DATA')
       OPEN(UNIT = 24,FILE = 'M:\FORTRAN\DATA\RR11.DATA')
       OPEN(UNIT = 25,FILE = 'M:\FORTRAN\DATA\RR22.DATA')
       OPEN(UNIT = 26,FILE = 'M:\FORTRAN\DATA\LEA.DATA')
       OPEN(UNIT = 27,FILE = 'M:\FORTRAN\DATA\S0.DATA')
       OPEN(UNIT = 28,FILE = 'M:\FORTRAN\DATA\K.DATA')
       OPEN(UNIT = S0FILE,FILE = 'M:\FORTRAN\DATA\S0LOOP.DATA')

       OPEN(UNIT = S0FILE1,FILE = 'M:\FORTRAN\DATA\avals.DATA')


       OPEN(UNIT = 40,FILE = 'M:\FORTRAN\DATA\DEBUG.DATA')
       OPEN(UNIT = 41,FILE = 'M:\FORTRAN\DATA\DETAILS.DATA')
```

```
       OPEN(UNIT = 42,FILE = 'M:\FORTRAN\DATA\CAPTION.DATA')
C*********************************************************************
C
C       START LOOP TO INCREMENT WHOLE PROGRAM THROUGH RANGE VALUES   10/18/2002
C
C*********************************************************************
C
C       START PROGRAM AND INITIALIZE CERTAIN VARIABLES
C
C*********************************************************************

       done = .FALSE.

c      SGGM0  = RE * SGA0
c      SGALF0 = (RE/2.00D0) * SGB0


C*********************************************************************
C
C       COMPUTE GAMMA TRUE AND ALFA TRUE FOR LATER COMPARISON
C
C*********************************************************************
       GMTRU = ATRUE*RE
       ALFTRU =(0.5)*BTRUE*RE

C*********************************************************************
C
C       GENERATE TRANSITION MATRIX PHI
C       IT WILL BE THE IDENTITY MATRIX FOR BOTH FILTERS
C
C*********************************************************************

       PHI(1,1) = 1.0D0
       PHI(2,2) = 1.0D0
       PHI(2,1) = 0.0D0
       PHI(1,2) = 0.0D0

C*********************************************************************
C
C       INITIALIZE FLAT EARTH FILTER STATE VECTOR AND COVARIANCE MATRIX
C
C*********************************************************************

       DO 20 I = 1,2
       DO 20 J = 1,2
       PF(I,J) = 0.0D0
 20      CONTINUE
       PF(1,1) = SGA0**2
       PF(2,2) = SGB0**2

C      DATA A0,B0/4.00E-4,0.15E-3/

       XF(1) = A0  * 1.0
       XF(2) = B0  * 1.0

C      ERRAF = XF(1) - ATRUE
C      ERRBF = XF(2) - BTRUE
```

Page 165

```
C*********************************************************************
C
C         INITIALIZE RATE LIMITING MATRIX FOR FLAT EARTH FILTER (Process Noise)
C
C*********************************************************************

      DO 21 I = 1,2
      DO 21 J = 1,2
      QF(I,J) = 0.0
 21      CONTINUE

!1    QF(1,1) = 1.0E-7
C     QF(1,1) = 1.0E-9
!2    QF(2,2) = 1.0E-9


c


C  ****************************************************
c     Range 1 to 3= 9 km, theta0 = 3 degrees SCENARIO:  Short/Low, in medium
altitude
c
c     QF(1,1) = 1.0E-9
c     QF(2,2) = 1.0E-9
C  ****************************************************
c     Range 1 to 3= 7 km, theta0 = 4 degrees SCENARIO:  Short/Low  low altitude
c
c     QF(1,1) = 1.0E-10
c     QF(2,2) = 1.0E-10
C  ****************************************************
c     Range 1 to 3= 9 km, theta0 = 32 degrees SCENARIO: Short/Medium
c
      QF(1,1) = 1.0E-7

      QF(2,2) = 1.0E-9
C  ****************************************************
c     Range 1 to 3= 9 km, theta0 = 32 degrees SCENARIO: Short/High
c
c     QF(1,1) = 1.0E-7
c     QF(2,2) = 1.0E-9
C  ****************************************************
c     Range 1 to 3= 57 km, theta0 = 5 degrees SCENARIO:  MEDIUM/LOW
c
c     QF(1,1) = 1.0E-10
c     QF(2,2) = 1.0E-10
C  ****************************************************
c     Range 1 to 3= 60 km, theta0 = -10 degrees SCENARIO: Medium/Medium
cc
c     QF(1,1) = 1.0E-7
c     QF(2,2) = 1.0E-9
C  ****************************************************
c     Range 1 to 3= 54 km, theta0 = 83 degrees SCENARIO:Medium/High
c
c     QF(1,1) = 1.0E-4
c     QF(2,2) = 1.0E-4
C  ****************************************************
c     Range 1 to 3= 202 km, theta0 = 0.20 degrees SCENARIO:Long/Low
```

Page 166

```
c
c     QF(1,1) = 1.0E-7
c     QF(2,2) = 1.0E-9
C ******************************************************
c     Range 1 to 3= 202 km, theta0 = 9 degrees SCENARIO: Long/Higher
c
c     QF(1,1) = 1.0E-7
c     QF(2,2) = 1.0E-9
C ******************************************************
c     Range 1 to 3= 450 km, theta0 = 3 degrees SCENARIO: Very Long/Low
c
c     QF(1,1) = 1.0E-7
c     QF(2,2) = 1.0E-9
C ******************************************************
c     Range 1 to 3= 450 km, theta0 = 3 degrees SCENARIO: Very Long/Medium angle
c
C*********************************************************************
C
C         INITIALIZE FLAT EARTH OBSERVATION VARIANCE MATRIX
C
C*********************************************************************

      DO 22 I = 1,2
      DO 22 J = 1,2  ·
      RF(I,J) = 0.0
 22      CONTINUE

 !    RF(1,1) = 1.0E+0
 !    RF(2,2) = 1.0E+0

c
C ******************************************************
c     Range 1 to 3= 9 km, theta0 = 3 degrees SCENARIO:  Short/Low, in medium
altitude
c
c     RF(1,1) = 1.0E-10
c     RF(2,2) = 1.0E-18
C ******************************************************
c     Range 1 to 3= 7 km, theta0 = 4 degrees SCENARIO:  Short/Low  low altitude
c
c     RF(1,1) = 1.0E-12
c     RF(2,2) = 1.0E-12
C ******************************************************
c     Range 1 to 3= 9 km, theta0 = 32 degrees SCENARIO: Short/Medium
c
      RF(1,1) = 1.0E-8
      RF(2,2) = 1.0E-8
C ******************************************************
c     Range 1 to 3= 9 km, theta0 = 32 degrees SCENARIO: Short/High
c
c     RF(1,1) = 1.0E-12
c     RF(2,2) = 1.0E-12
C ******************************************************
c     Range 1 to 3= 57 km, theta0 = 5 degrees SCENARIO:  MEDIUM/LOW
c
c     RF(1,1) = 1.0E-12
c     RF(2,2) = 1.0E-12
```

```
C ****************************************************
c      Range 1 to 3= 60 km, theta0 = -10 degrees SCENARIO: Medium/Medium
cc
c      RF(1,1) = 1.0E-8
c      RF(2,2) = 1.0E-8
C ****************************************************
c      Range 1 to 3= 54 km, theta0 = 83 degrees SCENARIO:Medium/High
c
c      RF(1,1) = 1.0E-8
c      RF(2,2) = 1.0E-8
C ****************************************************
c      Range 1 to 3= 202 km, theta0 = 0.20 degrees SCENARIO:Long/Low
c
c      RF(1,1) = 1.0E-8
c      RF(2,2) = 1.0E-8
C ****************************************************
c      Range 1 to 3= 202 km, theta0 = 9 degrees SCENARIO: Long/Higher
c
c      RF(1,1) = 1.0E-8
c      RF(2,2) = 1.0E-8
C ****************************************************
c      Range 1 to 3= 450 km, theta0 = 3 degrees SCENARIO: Very Long/Low
c
c      RF(1,1) = 1.0E-8
c      RF(2,2) = 1.0E-8
C ****************************************************
c      Range 1 to 3= 450 km, theta0 = 3 degrees SCENARIO: Very Long/Medium angle
c
C***********************************************************************
C
C        INITIALIZE RAY MODEL KALMAN FILTER STATE AND COVARIANCE
C
C***********************************************************************

      DO 30 I = 1,3
      DO 30 J = 1,3
      PR(I,J) = 0.0D0
 30     CONTINUE

C ****************************************************
c      DEFAULT
c
      PR(1,1) = SGGM0**2
      PR(2,2) = SGALF0**2
      PR(3,3) = SGK0**2

C     DATA A0,B0/400.0E-6,0.15E-3/
c     GAM0  =   A0 * RE
c     ALF0  =   (B0 * RE)/2.00D0

      K0 = 1.00D0

      XR(1) =  GAM0
      XR(2) =  ALF0
      XR(3) =  K0
```

```
C*********************************************************************
C
C         INITIALIZE RATE LIMITING MATRIX FOR RAY FILTER (Process Noise)
C
C*********************************************************************

          DO 31 I = 1,3
          DO 31 J = 1,3

          QR(I,J) = 0.0D0
 31       CONTINUE

!         QR(1,1) = 1.0E+0
!         QR(2,2) = 1.0E+0
!         QR(3,3) = 1.0E-12
c         QR(1,1) = 1.0E+3
c         QR(2,2) = 1.0E+3
c         QR(3,3) = 1.0E-1
c
C ********************************************************
c     Range 1 to 3= 9 km, theta0 = 3 degrees SCENARIO:  Short/Low, in medium
altitude
c
c         QR(1,1) = 1.0E+0
c         QR(2,2) = 1.0E+1
c         QR(3,3) = 1.0E+1
C ********************************************************
c     Range 1 to 3= 7 km, theta0 = 4 degrees SCENARIO:  Short/Low  low altitude
c
c         QR(1,1) = 1.0E+0
c         QR(2,2) = 1.0E+1
c         QR(3,3) = 1.0E+1
C ********************************************************
c     Range 1 to 3= 9 km, theta0 = 32 degrees SCENARIO: Short/Medium
c
          QR(1,1) = 1.0E+0
          QR(2,2) = 1.0E+0
          QR(3,3) = 1.0E+0
C ********************************************************
c     Range 1 to 3= 9 km, theta0 = 32 degrees SCENARIO: Short/High
c         QR(1,1) = 1.0E+0
c         QR(2,2) = 1.0E+0
c         QR(3,3) = 1.0E+8
C ********************************************************
c     Range 1 to 3= 57 km, theta0 = 5 degrees SCENARIO:  MEDIUM/LOW
c
c         QR(1,1) = 1.0E+0
c         QR(2,2) = 1.0E+1
c         QR(3,3) = 1.0E+1
C ********************************************************
c     Range 1 to 3= 60 km, theta0 = -10 degrees SCENARIO:Medium/Medium
c
c         QR(1,1) = 1.0E+0
c         QR(2,2) = 1.0E+0
c         QR(3,3) = 1.0E+8
C ********************************************************
c     Range 1 to 3= 54 km, theta0 = 83 degrees SCENARIO:Medium/High
```

```
c
c       QR(1,1) = 1.0E+0
c       QR(2,2) = 1.0E+0
c       QR(3,3) = 1.0E+0
C ********************************************************
c       Range 1 to 3= 202 km, theta0 = 0.20 degrees SCENARIO:Long/Low
c
c       QR(1,1) = 1.0E+0
c       QR(2,2) = 1.0E+0
c       QR(3,3) = 1.0E-2
C ********************************************************
c       Range 1 to 3= 202 km, theta0 = 9 degrees SCENARIO: Long/Higher
c
c       QR(1,1) = 1.0E+0
c       QR(2,2) = 1.0E+0
c       QR(3,3) = 1.0E-2
C ********************************************************
c       Range 1 to 3= 450 km, theta0 = 3 degrees SCENARIO: Very Long/Low
c
c       QR(1,1) = 1.0E+0
c       QR(2,2) = 1.0E+0
c       QR(3,3) = 1.0E-2
C ********************************************************
c       Range 1 to 3= 450 km, theta0 = 3 degrees SCENARIO: Very Long/Medium angle
c
C*****************************************************************************
C
C        INITIALIZE RAY MODEL OBSERVATION VARIANCE
C
C*****************************************************************************

        DO 32 I = 1,3
        DO 32 J = 1,3
        RR(I,J) = 0.0
 32        CONTINUE

c       RR(1,1) = 1.0E-1
c       RR(2,2) = 1.0E-1
c       RR(3,3) = 1.0E-4

!       RR(1,1) = 1.0E+3
!       RR(2,2) = 1.0E+3
!       RR(3,3) = 1.0E-14
c
C ********************************************************
c       Range 1 to 3= 9 km, theta0 = 3 degrees SCENARIO:  Short/Low, in medium
altitude
c
c       RR(1,1) = 1.0E+1
c       RR(2,2) = 1.0E-0
c       RR(3,3) = 1.0E-12
C ********************************************************
c       Range 1 to 3= 7 km, theta0 = 4 degrees SCENARIO:  Short/Low  low altitude
c
c       RR(1,1) = 1.0E+1
c       RR(2,2) = 1.0E-0
c       RR(3,3) = 1.0E-12
```

```
C ****************************************************
c     Range 1 to 3= 9 km, theta0 = 10 degrees SCENARIO: Short/Medium
c
      RR(1,1) = 1.0E-0
      RR(2,2) = 1.0E-0
      RR(3,3) = 1.0E+4
C ****************************************************
c     Range 1 to 3= 9 km, theta0 = 32 degrees SCENARIO: Short/High
c     RR(1,1) = 1.0E-6
c     RR(2,2) = 1.0E-6
c     RR(3,3) = 1.0E-2
C ****************************************************
c     Range 1 to 3= 57 km, theta0 = 5 degrees SCENARIO:  MEDIUM/LOW
c
c     RR(1,1) = 1.0E+1
c     RR(2,2) = 1.0E-0
c     RR(3,3) = 1.0E-12
C ****************************************************
c     Range 1 to 3= 60 km, theta0 = -10 degrees SCENARIO: Medium/Medium
c
c     RR(1,1) = 1.0E-3
c     RR(2,2) = 1.0E-6
c     RR(3,3) = 1.0E-2
C ****************************************************
c     Range 1 to 3= 60 km, theta0 = 40 degrees SCENARIO:Medium/High
c
c     RR(1,1) = 1.0E+0
c     RR(2,2) = 1.0E+0
c     RR(3,3) = 1.0E-0
C ****************************************************
c     Range 1 to 3= 202 km, theta0 = 0.20 degrees SCENARIO:Long/Low
c
c     RR(1,1) = 1.0E+8
c     RR(2,2) = 1.0E+8
c     RR(3,3) = 1.0E+0
C ****************************************************
c     Range 1 to 3= 202 km, theta0 = 9 degrees SCENARIO: Long/Higher
c
c     RR(1,1) = 1.0E+8
c     RR(2,2) = 1.0E+8
c     RR(3,3) = 1.0E+0
C ****************************************************
c     Range 1 to 3= 450 km, theta0 = 3 degrees SCENARIO: Very Long/Low
c
c     RR(1,1) = 1.0E+0
c     RR(2,2) = 1.0E+0
c     RR(3,3) = 1.0E+0
C ****************************************************
c     Range 1 to 3= 450 km, theta0 = 3 degrees SCENARIO: Very Long/Medium angle
c
C*********************************************************************
C
C        INITIALIZE EXECUTION LOG FOR SIMULATION ON UNITS 40,41
C           AND CALCULATE RANGE FROM EACH A/C to Reference A/C #1
C
C*********************************************************************
      If(debug) Then
```

```
      CALL FDATE (timest)
      DO 908 NFILE = 40,41
      WRITE(NFILE,700)
700     FORMAT(1H ,'********SIMULATION EXECUTION LOG********',//)
      WRITE(NFILE,701)
701     FORMAT(1H ,'-------INITIAL POSITION OF L-16 MEMBERS',/)
      WRITE(NFILE,*) timest

      IF (MDL .EQ. 1) THEN
            WRITE(NFILE,*) ' EXPONENTIAL MODEL RESULTS FOR A/C', ACNUM
      ELSE IF (MDL .EQ. 2 ) THEN
            WRITE(NFILE,*) ' SMITH-WEINTRAUB MODEL RESULTS FOR A/C', ACNUM
      ELSE
            WRITE(NFILE,*) ' THAYER MODEL RESULTS FOR A/C', ACNUM
      ENDIF

      DO 35 I = 1,4
      RNGX = ( (H(I)+RE) * DCOS(LAT(I)/RAD) * DCOS(LON(I)/RAD)   )
     1 - ( (H(1)+RE) * DCOS(LAT(1)/RAD) * DCOS(LON(1)/RAD) )
      RNGY = ( (H(I)+RE) * DCOS(LAT(I)/RAD) * DSIN(LON(I)/RAD)   )
     1 - ( (H(1)+RE) * DCOS(LAT(1)/RAD) * DSIN(LON(1)/RAD) )
      RNGZ = ( (H(I)+RE) * DSIN(LAT(I)/RAD) )
     1      - ( (H(1)+RE) * DSIN(LAT(1)/RAD) )

      RNG = RNGX * RNGX + RNGY * RNGY + RNGZ * RNGZ

      RNG = DSQRT(RNG)

      IF ( I .NE. 1) THEN
      ELEV = DATAN( (H(ACNUM) - H(1) )/RNG)
      write(40,915) I,ELEV*RAD
      write(41,915) I,ELEV*RAD
      ENDIF

      IF(I .EQ. ACNUM) then
      RNG3 = RNG
      ENDIF
      WRITE(NFILE,702)I,LAT(I),LON(I),H(I)
702     FORMAT(1H ,'A/C #',1X,I2,1X,'LAT=',1X,F12.4,1X,'LON=',1X,
     1          F12.4,1X,'HGT=',1X,F12.2)
      IF (I .EQ. 1) THEN
      GO TO 35
      ELSE
      WRITE(NFILE,727)I,RNG/1000.0D0, RNG/1852.0D0 ! range in km and nm

727     FORMAT(1H ,'A/C #',1X,I2,1X,'Range to A/C #1 =',
     1  1X,E12.6,' km  = ', 1X, E12.6,' nm ')
      ENDIF
 35     CONTINUE
      done = .TRUE.


      WRITE(NFILE,703)
703     FORMAT(1H ,/,'--------TIMESTEP AND SIMULATION DURATION',/)
      WRITE(NFILE,704)DELT, TIMEF
704     FORMAT(1H ,'DELT = ',1X,F5.2,1X,'TIMEF = ',1X,F5.2,/)
```

```
      WRITE(NFILE,705)
705     FORMAT(1H ,'--------TRUE ATMOSPHERIC PARAMETERS',/)
      WRITE(NFILE,706)ATRUE,BTRUE
706     FORMAT(1H ,'ATRUE = ',1X,E12.6,2X,'BTRUE = ',1X,E12.6,/)
      WRITE(NFILE,907)ATRUE*RE,(0.5)*BTRUE*RE
907     FORMAT(1H ,'GMTRU = ',1X,E12.6,2X,'ALFTRU = ',1X,E12.6,/)
      WRITE(NFILE,707)
707     FORMAT(1H ///)
908   END DO

      Else

      DO 36 I = 1,4
      RNGX = ((H(I)+RE) * DCOS(LAT(I)/RAD) * DCOS(LON(I)/RAD)   )
     1 - ((H(1)+RE) * DCOS(LAT(1)/RAD) * DCOS(LON(1)/RAD) )
      RNGY = ((H(I)+RE) * DCOS(LAT(I)/RAD) * DSIN(LON(I)/RAD)   )
     1 - ((H(1)+RE) * DCOS(LAT(1)/RAD) * DSIN(LON(1)/RAD) )
      RNGZ = ((H(I)+RE) * DSIN(LAT(I)/RAD) )
     1       - ((H(1)+RE) * DSIN(LAT(1)/RAD) )


      RNG = RNGX * RNGX + RNGY * RNGY + RNGZ * RNGZ

      RNG = DSQRT(RNG)

      ELEV = DATAN( (H(ACNUM) - H(1) )/RNG)
      write(40,915) I,ELEV*RAD
      write(41,915) I,ELEV*RAD


      IF(I .EQ. ACNUM) then
      RNG3 = RNG
      ENDIF

  36  Continue


      Endif  ! if (debug) logic
915   FORMAT(1X,'L16 member 1 to A/C No. ', I3': angle (deg)= ',
     1 E12.4)

      IF (MDL .EQ. 1) THEN
           WRITE(42,*) ' EXPONENTIAL MODEL RESULTS '
      ELSE IF (MDL .EQ. 2 ) THEN
           WRITE(42,*) ' SMITH-WEINTRAUB MODEL RESULTS '
      ELSE
           WRITE(42,*) ' THAYER MODEL RESULTS '
      ENDIF
      WRITE(42,730) LAT(1),LON(1),H(1),LAT(ACNUM),
     1 LON(ACNUM),H(ACNUM),RNG3/1000.0D0,RNG3/1852.0D0
730     FORMAT(1H ,'RECEIVER',1X,'LAT=',1X,F10.4,1X,'deg, LON=',1X,
     1        F10.4,1X,'deg, HGT=',1X,F10.2,' m',/,
     2        1H ,'EMITTER',1X,' LAT=',1X,F10.4,1X,'deg, LON=',1X,
     3        F10.4,1X,'deg, HGT=',1X,F10.2,' m',/,
     4  1H ,'RANGE = ',F10.4,' KM ',F10.4,' NM ')

      If(debug) Then
      WRITE(27,921) ACNUM, K
```

```
 921       FORMAT(2X,'TIME ',2X,'  S0 ',3X,' S0EST ',5X,'S0ERR ',5X,
     1' KC * YR(1)',3X,' LEA',5X,'L16 ERR',2X,'ACNUM',I3,' K =',F6.3)
          Endif

C*******************************************************************
C
C         INITIALIZE AIRCRAFT ANGLE VARIABLES
C         ANGLES MUST BE EXPRESSED IN RADIANS
C         ALTITUDES ARE EXPRESSED IN METERS
C
C*******************************************************************

          DO 40 I = 1,4
          LAT(I) = LAT(I)/RAD
          LON(I) = LON(I)/RAD
  40      CONTINUE

C*******************************************************************
C
C         MAIN PROGRAM TIME LOOP STARTS HERE
C
C         LOOP APPLIES RANGE MEASUREMENTS FROM A/C 2,3,AND 4 IN TURN
C         TO A/C 1.  THESE MEASUREMENTS ARE APPLIED TO TWO COMPARATIVE
C         KALMAN KALMAN FILTERS.  THE FIRST IS A FLAT EARTH (FE) MODEL.
C         THE SECOND UTILIZES A SPHERICAL EARTH (RE) MODEL. TIME IS
C         INCREMENTED PRIOR TO EACH (SCALAR) OBSERVATION IN EITHER
C         FILTER
C
C*******************************************************************

 100      CONTINUE

          If(debug) Then
          WRITE(40,708) TIME
 708      FORMAT(1H ,'*****BEGINNING TIME CYCLE AT',1X,F6.1,'SEC',/)
          Endif

          done = .FALSE.

          DO 200 I = 2,4
          PARAMS(29,I)= TIME
          TIME = TIME + DELT

C*******************************************************************
C
C         FOR EACH INDEXED AIRCRAFT, INVOKE TRUE WORLD RANGE MODEL
C         BY CALLING SUBROUTINE TWGEN(TRUE WORLD GENERATION)
C
C*******************************************************************

          CALL TWGEN(ATRUE,BTRUE,LAT(1),LON(1),H(1),LAT(I),LON(I),H(I),
     1               S0,SE,PHI0,LE,LC,THETA0,TCOM,RE,PI,MDL)

          If(I.EQ.ACNUM) Then
          S0SAVED = S0
          THETA03 = THETA0
          Endif
```

```
      If(debug) Then
      WRITE(40,709)I
 709     FORMAT(1H ,'--------TWGEN PARAMETERS FOR A/C  ',I2/)
      WRITE(40,710)PHI0,S0,SE
 710     FORMAT(1H ,'PHI0/S0/SE',3(2X,E12.6))
      WRITE(40,711)LE,LC,THETA0,TCOM,THETA0*RAD,TCOM*RAD
 711     FORMAT(1H ,'LE/LC/THETA0(DEG)/TCOM(DEG)/THETA0(DEG)/
     1TCOM(DEG)',6(2X,E12.6)//)
      Endif

      PARAMS(1,I)=PHI0
      PARAMS(2,I)=S0
      PARAMS(3,I)=SE
      PARAMS(4,I)=LE
      PARAMS(5,I)=LC

      PARAMS(6,I)=THETA0
      PARAMS(7,I)=TCOM

C*****************************************************************
C
C     COMPUTE MEASURED RANGE TO TRANSMITTING PLATFORM IN 12.5 NS
C     COUNTS.  YF(1),YR(1) EQUAL TAUM IN 12.5 NS COUNTS.  TRUE
C     WORLD MODEL IN TWGEN PROVIDES SE FOR MEASUREMENT EQUATION
C
C*****************************************************************

       TAUM = FLOAT(INT(SE/KC))

      YF(1)= TAUM
      YF(2)= 0.0
      YR(1)= TAUM
      YR(2)= 0.0
      YR(3)= 0.0

      If(debug) Then
      WRITE(40,712) TAUM
 712     FORMAT(1H ,'--------MEASURED RANGE IN COUNTS = ',E12.6/)
      Endif

      PARAMS(8,I)=TAUM
C*****************************************************************
C
C     GENERATE HF MATRIX FOR INDEXED PLATFORM
C
C*****************************************************************

       CALL HFMATR(HF,XF,H,S0,KC,I)

      If(debug) Then
       WRITE(40,713) HF(1,1),HF(1,2)
 713      FORMAT(1H ,'---------HF11/HF12---',2(2X,E12.6)/)
      Endif

      PARAMS(9,I) =HF(1,1)
      PARAMS(10,I)=HF(1,2)
```

Page 175

```
C*********************************************************************
C
C       COMPUTE PREDICTED OBSERVATION FOR FE FILTER
C
C*********************************************************************

      YFEXP(1) = (S0/KC)/(1.0 - XF(1) + 0.5*XF(1)*XF(2)*
     1                      (H(I)+H(1)))
      YFEXP(2) = 0.0

      RSL = YF(1) - YFEXP(1)

      If(debug) Then
            IF(I.EQ.ACNUM)THEN
      WRITE(17,*)TIME,RSL
      endif
      WRITE(40,714)YF(1),YFEXP(1),RSL
 714     FORMAT(1H ,'--------YF1/YEXP1/RSL---',3(2X,E12.6)/)
      Endif

      PARAMS(11,I)=YF(1)
      PARAMS(12,I)=YFEXP(1)
      PARAMS(13,I)=RSL
C*********************************************************************
C
C       INVOKE FE FILTER KALMAN PROCESSING
C
C*********************************************************************

       CALL FILTR9(PHI,HF,PF,QF,RF,XF,YF,YFEXP)

      If(debug) Then
       WRITE(40,715)1,1,2,2,PF(1,1),PF(2,2)
       WRITE(40,715)1,2,2,1,PF(1,2),PF(2,1)
 715     FORMAT(1H ,'--------PF',I1,I1,'/PF',I1,I1,'---',2(2X,E12.6)/)
      Endif

      PARAMS(14,I)=PF(1,1)
      PARAMS(15,I)=PF(2,2)
      PARAMS(34,I)=XF(1)
C*********************************************************************
C
C       RECORD FE FILTER ERROR AND COVARIANCE VALUES
C
C*********************************************************************
      If(debug) Then

      IF(I.EQ.ACNUM)THEN
      SIG = DSQRT(PF(1,1))
      WRITE(11,*)TIME,SIG

      SIG = DSQRT(PF(2,2))

      WRITE(12,*)TIME,SIG
      WRITE( 9,*)TIME,XF(1)
      WRITE(10,*)TIME,XF(2)
```

```
c       WRITE(24,*)TIME,QF(1,1)
c       WRITE(25,*)TIME,QF(2,2)
        endif
        Endif
C*********************************************************************
C
C       COMPUTE S0 STANDARD DEVIATION FOR A/C 2 MESSAGES (FE)
C
C*********************************************************************

        IF(I.EQ.ACNUM)THEN
           DCDA = KC*TAUM*(-1.0 + 0.5*XF(2)*(H(1) + H(I)))
           DCDB = 0.5*KC*TAUM*(XF(1)*(H(1) + H(I)))

           SIG2 = DCDA*DCDA*PF(1,1) + DCDB*DCDB*PF(2,2)
     1              + 2.*DCDA*DCDB*PF(1,2)
           SIG = DSQRT(SIG2)
           SGS0FE = SIG

        If(debug) Then

           WRITE(15,*) TIME,SIG
        Endif

        ENDIF


C*********************************************************************
C
C       COMPUTE OBSERVATION FOR RE FILTER
C
C*********************************************************************

c       Original LEA:  compute PHIEST as SE (= KC*TAUM)/RE???
        PHIEST = SE/RE   ! 10/30/02

        LEA = XR(3)*XR(1)*DEXP(-2.*XR(2)*(TCOM*TCOM + H(1)/RE))*PHIEST

c       LEA = DEXP(-2.0*(H(1)/RE)* XR(2)) *
c     1 XR(1) * ((TCOM + (PHIEST/XR(3)))/(XR(3) * XR(2)))
C Preceeding is new LEA for 10/4/02

c       LEA = XR(3)*XR(1)*DEXP(-2.*XR(2)*(TCOM*TCOM + H(1)/RE))*PHIEST
c       LEA = 0.5*DSQRT(PI)*XR(1)*DEXP(1.0D0*XR(2)*(TCOM*TCOM) - H(1))
c     1       * (DERF(DSQRT(XR(2)/XR(3))* (PHIEST + XR(3)* TCOM)) -
c     2           DERF(DSQRT(XR(2)/XR(3))* XR(3)*TCOM)   )
c Preceeding is exact LEA

c       LEA1 = 0.5
c       LEA2 = DSQRT(PI)
c       LEA3 = XR(1)
c       LEA4 = DEXP(1.0D0*XR(2)*(TCOM*TCOM) - H(1) )
c          LEA5 = (   DERF(DSQRT(XR(2)/XR(3))* (PHIEST + XR(3)* TCOM)) -
c     1           DERF(DSQRT(XR(2)/XR(3))* XR(3)*TCOM)   )
c        LEA= LEA1 * LEA2 * LEA3 *LEA4 * LEA5
         B = 2.0*XR(2)/RE
!!         LCA = (S0*(B*LEA)**2)/24.0  ! Should S0 be SE (= KC*TAUM) ?
```

```fortran
c       LCA = (KC*TAUM*(B*LEA)**2)/24.0

        If(debug) Then

        WRITE(40,556)XR(3),XR(1),XR(2),TCOM
  556   FORMAT(1H ,'&&&&K(XR(3))/XR1/XR2/TCOM',4(1X,E12.6))

        WRITE(40,716)LEA,LCA
  716   FORMAT(1H ,'--------LEA/LCA---',2(2X,E12.6)/)

        Endif   ! Debug

c            IF(I.EQ.ACNUM)THEN
c            WRITE(26,919)TIME,LEA,LEA3,LEA4,LEA5
c 919      FORMAT(1X,'TIME ',E12.6,' LEA ',E12.6,
c     2' LEA3 (XR(1)) ',E12.6,/,' LEA4 ',E12.6,' LEA5 ',E12.6,//)
c            ENDIF

c       TAUM = (LEA + LCA + S0)/KC
            TAUM = (LEA + S0)/KC
        PARAMS(16,I)=XR(3)
        PARAMS(17,I)=XR(1)
        PARAMS(18,I)=XR(2)
        PARAMS(19,I)=TCOM
        PARAMS(20,I)=LEA
        PARAMS(21,I)=LCA


        YREXP(1) = TAUM
        YREXP(2) = 0.0

        RSL = YR(1) - YREXP(1)

        If(debug) Then
            IF(I.EQ.ACNUM)THEN
        WRITE(18,*) TIME,RSL
        endif
        WRITE(40,555)YR(1),YREXP(1)
  555     FORMAT(1H ,'&&&&YR1/YREXP1',2(2X,E12.6))

        WRITE(40,717)RSL
  717     FORMAT(1H ,'--------RSL----',1X,E12.6/)

        Endif

        PARAMS(22,I)=YR(1)
        PARAMS(23,I)=YREXP(1)
        PARAMS(24,I)=RSL
C******************************************************************
C .
C       GENERATE HR OBSERVATION MATRIX FOR RE FILTER
C
C******************************************************************

        CALL HRMATR(HR,XR,H,TCOM,PHI0,KC,RE,SE)

C******************************************************************
```

```
C
C         INVOKE RE FILTER KALMAN PROCESSING
C
C******************************************************************

          CALL REFLTR(PHI,HR,PR,QR,RR,XR,YR,YREXP)

        If(debug) Then
         WRITE(40,718)1,1,PR(1,1)
         WRITE(40,718)2,2,PR(2,2)
         WRITE(40,718)3,3,PR(3,3)
  718       FORMAT(1H ,'--------PR',I1,I1,'---',2X,E12.6/)
         WRITE(40,719)HR(1,1),HR(1,2),HR(1,3)
  719       FORMAT(1H ,'--------HR11/HR12/HR13----',3(2X,E12.6)/)
        Endif

        PARAMS(25,I)=PR(1,2)
        PARAMS(26,I)=PR(2,1)
        PARAMS(27,I)=HR(1,1)
        PARAMS(28,I)=HR(1,2)

        If(debug) Then
              IF(I.EQ.ACNUM)THEN
        WRITE(24,*)TIME,RR(1,1)
        WRITE(25,*)TIME,RR(2,2)
        endif
        Endif
C******************************************************************
C
C        RECORD RE FILTER STATE AND COVARIANCE VALUES
C
C******************************************************************

        SIG = DSQRT(PR(1,1))

        If(debug) Then

        IF(I.EQ.ACNUM)THEN
        WRITE(19,*) TIME,SIG

        SIG = DSQRT(PR(2,2))

        WRITE(20,*) TIME,SIG
        WRITE(13,*)TIME,XR(1)/RE !, (DEXP(XR(2)*2.0D0*H(ACNUM)/RE))*
  !     1( (1.0D0-(1.0D0/XR(3) ) ) )/ (4.0*XR(2)*2.0D0) ! a from gamma, a from
gamma & k
        WRITE(14,*) TIME,XR(2)*2.0/RE   ! b from alpha

        endif
        Endif


C******************************************************************
C
C        COMPUTE S0 STANDARD DEVIATION FOR RE FILTER
C        AND S0 ERRORS FOR FE AND RE FILTERS
C******************************************************************
```

```
        IF(I.EQ.ACNUM)THEN
c original model
        PHIEST = SE/RE   ! 10/30/02
            FAC = -2.0*(TCOM*TCOM + H(1)/RE)
         .  DLDG = XR(3)*DEXP(XR(2)*FAC)*PHIEST
            DLDA = XR(3)*XR(1)*FAC*DEXP(XR(2)*FAC)*PHIEST
c new model
c        FAC = DEXP(-2.0*(H(1)/RE)* XR(2)) *
c    1 ((TCOM + (PHIEST/XR(3)))/(XR(3) * XR(2)))
c     DLDG = FAC
c     DLDA = FAC * ( (-2.0 + H(1))/RE - (1.0/XR(2)))

        SIG2 = DLDG*DLDG*PR(1,1) + DLDA*DLDA*PR(2,2)
     1            + 2.0*DLDG*DLDA*PR(1,2)
        SIG = DSQRT(SIG2)

      If(debug) Then
          WRITE(16,*)TIME,SIG
      Endif

        SGS0RE = SIG

        S0EST = YF(1)*KC*(1.0 - XF(1) + 0.5*XF(1)*XF(2)*
     1                    (H(1) + H(I)))
        S0ERR = S0 - S0EST   ! FE Model

      oldaval = 2.95E-4
      oldab2 = 1.11893E-8
      oldbval = 0.80613E-4
      L16err1 = S0-(KC* YR(1))*(1.0D0 -oldaval+(oldab2*(H(1)+H(I)))) !current
L16 model

      L16FACTR = (oldaval/oldbval) *
     1 ((DEXP(-oldbval*H(I)) - DEXP(-oldbval*H(1)) )/(H(I)-H(1)) )
      L16err2 = S0 - (KC * YR(1)/ (1.0D0  - L16FACTR) )

c      L16FACTR = 0.0
c      L16err2 = 0.0

      If(debug) Then
      WRITE(21,950)TIME,S0ERR,L16err1,L16err2
 950  FORMAT(F8.2,F20.6,F20.6,F20.6)
      Endif

        PARAMS(30,I)= S0ERR   ! FE Model

        S0EST = KC*YR(1) - LEA   !RE Model
        S0ERR = S0 - S0EST


c     The current Link-16 compensation model for the range S0 calculation is:
c   S0 = c *  TOA * [ 1 - a + ab/2 (hemit + hrcvr)],
c where a =  2.95 * 10 -4 and ab/2 = 1.1893 * 10-8 /meter, the receiver at is at
altitude hrcvr and emitter at hemit.


      If(debug) Then
```

```
            WRITE(22,*)TIME,S0ERR
            WRITE(28,*)TIME,XR(3)
            WRITE(27,920)TIME,S0,S0EST,S0ERR,KC * YR(1),LEA

 920  FORMAT(F6.1,F10.1,F10.1,F10.1,3X,F10.3,F10.3,F10.3)
      Endif

      PARAMS(31,I)= S0ERR    ! RE Model
      ENDIF

 200     CONTINUE

      If(debug) Then
      WRITE(40,707)

      WRITE(41,900)TIME-3*DELT
 900     FORMAT(1X,'*****BEGINNING TIME CYCLE AT',F10.4,//,3X,
     1  ' A/C Number ',4X,'2',12X,'3',12X,'4',/)
       DO 910 I=1,28
       WRITE(41,909)TYPE(I),PARAMS(I,2),
     1  PARAMS(I,3),PARAMS(I,4)
 909     FORMAT(1X,A,1X,E12.6,1X,E12.6,1X,E12.6,/)
 910     END DO

      WRITE(41,909)TYPE(30),PARAMS(30,2),
     1  PARAMS(30,3),PARAMS(30,4)
      WRITE(41,909)TYPE(31),PARAMS(31,2),
     1  PARAMS(31,3),PARAMS(31,4)
      WRITE(41,911)TYPE(29),PARAMS(29,2),
     1  PARAMS(29,3),PARAMS(29,4)
 911     FORMAT(1X,A,1X,F12.6,1X,F12.6,1X,F12.6,////)
      WRITE(41,770)1,1,RR(1,1),2,2,RR(2,2)
 770  FORMAT('RR(',I1,',',I1,') = ', E12.2,
     1'  RR(',I1,',',I1,') = ', E12.2)

      Endif   ! debug


       IF(TIME.LE.TIMEF)GO TO 100

       WRITE(S0FILE,922) H(1)/1000.D0,H(3)/1000.D0,RNG3/1000.D0,
     1 PARAMS(31,ACNUM),PARAMS(30,ACNUM),SGS0RE,SGS0FE,THETA03*RAD,
     2 PARAMS(16,ACNUM),PHIEST*RAD,PARAMS(17,ACNUM)

 922   FORMAT(2(F4.0),F6.1,4(E10.2),2F8.2,F6.2,1X,E8.2)
       WRITE(S0FILE1,923) PARAMS(11,ACNUM)/RE, PARAMS(34,ACNUM)
 923   FORMAT(1X,2E12.2)


      CLOSE(9, STATUS = 'KEEP' )
      CLOSE(10,STATUS = 'KEEP' )
      CLOSE(11,STATUS = 'KEEP' )
      CLOSE(12,STATUS = 'KEEP' )
      CLOSE(13,STATUS = 'KEEP' )
      CLOSE(14,STATUS = 'KEEP' )
      CLOSE(15,STATUS = 'KEEP' )
      CLOSE(16,STATUS = 'KEEP' )
```

```
        CLOSE(17,STATUS = 'KEEP' )
        CLOSE(17,STATUS = 'KEEP' )
        CLOSE(18,STATUS = 'KEEP' )
        CLOSE(19,STATUS = 'KEEP' )
        CLOSE(20,STATUS = 'KEEP' )
        CLOSE(21,STATUS = 'KEEP' )
        CLOSE(22,STATUS = 'KEEP' )
C       CLOSE(23,STATUS = 'KEEP' )
        CLOSE(24,STATUS = 'KEEP' )
        CLOSE(25,STATUS = 'KEEP' )
        CLOSE(26,STATUS = 'KEEP' )
        CLOSE(27,STATUS = 'KEEP' )
        CLOSE(28,STATUS = 'KEEP' )

        CLOSE(40,STATUS = 'KEEP' )
        CLOSE(41,STATUS = 'KEEP' )
        CLOSE(42,STATUS = 'KEEP' )
        CLOSE(S0FILE,STATUS = 'KEEP')
        CLOSE(S0FILE1,STATUS = 'KEEP')

        STOP
        END



        SUBROUTINE FILTR9(PHI,HX,PX,Q,R,X,Y,YEXP)
        IMPLICIT NONE
C*********************************************************************
C
C      GENERALIZED KALMAN FILTER EXECUTIVE ROUTINE
C
C*********************************************************************

        REAL*8 PHI(2,2),HX(2,2),PX(2,2),Q(2,2),R(2,2),X(2),Y(2),
     1          YEXP(2)

        REAL*8 G(2,2),DUM(2,2),DUM1(2,2),DENOM(2,2),DUM2(2,2),DUM3(2,2)
        REAL*8 DUMX(2,2),M(2,2),D
        INTEGER*4 L2(2),NW(2),I,J
        LOGICAL FLAG


C----------------------------------------------------------------------



C*********************************************************************
C
C      COMPUTE KALMAN GAIN MATRIX  G(2,2)
C
C*********************************************************************
C ADDED 9/23/02 RRD, assume transition matrix PHI = I
        PX(1,1) = PX(1,1) + Q(1,1)
        PX(2,2) = PX(2,2) + Q(2,2)
C END ADDED CODE

        CALL DTMAML(PX,HX,DUM,2,2,2,2,0,1)
        CALL DTMAML(HX,DUM,DENOM,2,2,2,2,0,0)
        DO 201 I = 1,2
        DO 201 J = 1,2
```

```fortran
          DUM1(I,J) = DENOM(I,J) + R(I,J)
  201     CONTINUE
          CALL DMINV1(DUM1,2,D)
          CALL DTMAML(DUM,DUM1,G,2,2,2,2,0,0)


C*********************************************************************
C
C        APPLY KALMAN STATE CORRECTION
C
C*********************************************************************

          X(1) = X(1) + G(1,1)*(Y(1) - YEXP(1))
          X(2) = X(2) + G(2,1)*(Y(1) - YEXP(1))


C*********************************************************************
C
C        COVARIANCE MATRIX PX OBERVATION UPDATE
C
C*********************************************************************

          CALL DTMAML(G,HX,DUM2,2,2,2,2,0,0)
          DO 20 I = 1,2
          DO 20 J = 1,2
          DUM2(I,J) = -DUM2(I,J)
  20      CONTINUE
          DO 25 I = 1,2
          DUM2(I,I) = 1.0 + DUM2(I,I)
  25      CONTINUE
          CALL DTMAML(DUM2,PX,DUM3,2,2,2,2,0,0)
          DO 30 I = 1,2
          DO 30 J = 1,2
          PX(I,J) = DUM3(I,J)
  30      CONTINUE


          RETURN
          END

          SUBROUTINE HFMATR(HF,XF,H,S0,KC,JJ)
          IMPLICIT NONE
C*********************************************************************
C
C        HFMATR GENERATES THE OBSERVATION MATRIX HF FOR THE
C        FLAT EARTH (FE) KALMAN FILTER
C
C*********************************************************************

          REAL*8 HF(2,2),XF(2),H(4),S0,KC
          REAL*8 FAC1,FAC2
          INTEGER*4 JJ,I,J

          FAC1 = (1.0 - 0.5*XF(2)*(H(JJ) + H(1)))    !changed first + to -
          FAC2 = (1.0 - XF(1) + 0.5*XF(1)*XF(2)*(H(JJ)+H(1)))

          DO 10 I = 1,2
```

```
      DO 10 J = 1,2
      HF(I,J) = 0.0
10      CONTINUE


      HF(1,1) = (S0/KC)*FAC1/FAC2**2
      HF(1,2) = -(S0/2.*KC)*(XF(1)*(H(JJ)+H(1)))/FAC2**2

      RETURN
      END


      SUBROUTINE HRMATR(HR,XR,H,TCOM,PHI0,KC,RE,SE)
      IMPLICIT NONE
C*******************************************************************
C
C      HRMATR GENERATES THE OBSERVATION MATRIX HR FOR THE
C      NEW RAY MODEL (RE) KALMAN FILTER of 10/07/02 with an added (third)
C       state, K   (RRD)
C
C*******************************************************************

      REAL*8 HR(3,3),XR(3),H(4),TCOM,PHI0,KC,RE,SE,PHIEST
      REAL*8 FAC,FAC1,FAC2,LEA,TAUM,LETERM

      INTEGER*4 I,J

      TAUM = SE/KC
      PHIEST = SE/RE   ! 10/30/02

      DO 10 I = 1,3
      DO 10 J = 1,3

      HR(I,J) = 0.0
10      CONTINUE


C      HRMATR (Joel)
C
C      XR(1) = gamma
C      XR(2) = alpha

      FAC       = -2.* (TCOM*TCOM + (H(1)/RE))
      FAC1      = FAC * XR(2)
      LETERM    = XR(3) * XR(1) * DEXP(FAC1)* PHIEST  ! = Le

      HR(1,1) = LETERM/(KC * XR(1))

      HR(1,2) = FAC * LETERM/KC

      HR(1,3) = LETERM/(KC * XR(3))


c      Correct the original approximation by adding LC derivative term to H
matrix, RRD 10/1/02
c      LEA = XR(1)* DEXP(FAC*XR(2))* PHIEST
c      HR(1,1) =  HR(1,1) + (TAUM/(3.0*RE*RE)) * (XR(2)*XR(2) *XR(1)) *
```

Page 184

```
c      1 LEA * DEXP(FAC*XR(2))*PHIEST
c       HR(1,2) =  HR(1,2) + (TAUM/(3.0*RE*RE)) * (XR(2)*LEA) *
c      1 (LEA + XR(2)*XR(1)*FAC*DEXP(FAC*XR(2))*PHIEST)


c      HRMATR (Orig. approx. 9/30/02)
c      HR(1,1) = (XR(3)/KC)*DEXP(XR(2)*(TCOM*TCOM - 2.0* H(1)/RE))*PHIEST
c      HR(1,2) = (XR(3)/KC)* (TCOM*TCOM - 2.0* H(1)/RE)*
c      1       DEXP(XR(2)*(TCOM*TCOM - 2.0* H(1)/RE))*PHIEST

c      HRMATR (Ralph approx. 10/4/02)  USE FOR MED/HIGH SCENARIO
c      FAC2 = DEXP(-2.0*(H(1)/RE)* XR(2)) *
c      1 ((TCOM + (PHIEST/XR(3)))/(XR(3) * XR(2)) )
c      HR(1,1) = FAC2/KC
c      HR(1,2) = (FAC2/KC) * ((-2.0 + H(1))/RE - (1.0/XR(2)) )
c      HR(1,3) = (-1.0D0/KC) * (XR(1)/(XR(2)*XR(3)*XR(3))) *
c      1 (DEXP(-2.0D0*XR(2)*(H(1)/RE))) * (TCOM + (PHI0/XR(3)) )


c      HR(1,3) = LETERM/(KC * XR(3))


c      HR(1,1) = HR(1,1) * 100.0

c      HR(1,2) = HR(1,2) * 100.0

c      HR(1,3) = HR(1,3) * 100.0



       RETURN
       END


       SUBROUTINE TWGEN(ATRUE,BTRUE,LAT1,LON1,H1,LAT2,LON2,H2,
      1                 S0,SE,PHI0,LE,LC,THETA0,TCOM,RE,PI,MODEL)
       IMPLICIT NONE
C****************************************************************
C
C      TWGEN GENERATES THE TRUE WORLD MODEL
C
C****************************************************************

       REAL*8 ATRUE,BTRUE,LAT1,LON1,H1,LAT2,LON2,H2,S0,SE,PHI0
       REAL*8 LE,LC,THETA0,RE,TCOM,PI

       REAL*8 SPHI2,THETA,EPSLN1,EPSLN2,PHI,N,DL
       REAL*8 DPHI,DH,HP1,NP1,DNDH,DTHETA,DS,DSE,HA,HB,DHDTH

       REAL*8 ALFA,C,FAC1,FAC2,FAC3,ERF2,H,S,RAD
       DATA RAD/57.2957795131/
       DATA EPSLN1/0.001/
       DATA EPSLN2/0.00/
       DATA THETA/0.000001/
       DATA DL/10.0/

       INTEGER*4 I,MODEL
```

```
C MODEL 1 = EXPONENTIAL, 2 = SMITH-WEINTRAUB, 3 = THAYER
      REAL*8 LVALUE,TEMP,TEMP0,TEMP1,P0,HREF,PRESS
      REAL*8 XVALUE,EHUMID,NFRACT
C******************************************************************
C
C      COMPUTE INCLUDED ANGLE PHI0 BETWEEN PLATFORMS
C      COMPUTE TRUE RANGE VALUE S0
C
C******************************************************************

      PHI0 = DACOS(DCOS(LAT1)*DCOS(LAT2)*(DCOS(LON1 - LON2)) +
     1               DSIN(LAT1)*DSIN(LAT2))


c     SPHI2 = DSIN(0.5*PHI0)**2
c     S0 = DSQRT((H2-H1)**2 + 4.0*(RE+H1)*(RE+H2)*SPHI2)


      S0 = DSQRT((RE+H1)**2 + (RE+H2)**2 -
     1       2.0*(RE+H1)*(RE+H2)*DCOS(PHI0))

C******************************************************************
C
C      COMPUTE INITIAL ESTIMATE FOR THETA0
C
C******************************************************************

      TCOM = DASIN((H2 - H1 - (S0**2/(2.*RE)))/S0)

c     write(*,*) "tcom = ", tcom

C******************************************************************
C
C      SOLVE RAY MODEL USING INITIAL ESTIMATE OF THETA0
C      CORRECT TO AGREE WITH H2 AT END POINT
C      OUTPUT TRUE SE,TRUE S, AND FINAL VALUE OF THETA0
C
C******************************************************************

      DO 100 I = 1,3
      IF(I.EQ.1)THETA = TCOM
      IF(I.EQ.2)THETA = TCOM + EPSLN1
      IF(I.EQ.3)THETA = TCOM + EPSLN2

      H = H1
      S = 0.0
      SE= 0.0
      PHI = 0.0

 800    CONTINUE

      GO TO (51,52) , MODEL

 51    CONTINUE

C******************************************************************
```

```
C
C        GENERATE INDEX OF REFRACTION USING EXPONENTIAL MODEL
C
C****************************************************************

       N = 1.0 + ATRUE*EXP(-BTRUE*H/1.)
       DPHI = DL/(RE + H)
       DH = (RE + H)*DTAN(THETA)*DPHI
       HP1 = H + DH
       NP1 = 1.0 + ATRUE*EXP(-BTRUE*HP1/1.)

       GO TO 55
  52   CONTINUE
C****************************************************************
C
C        GENERATE INDEX OF REFRACTION USING SMITH-WEINTRAUB MODEL
C
C****************************************************************

       LVALUE = 0.0065D0
       TEMP0 = 288.15D0   !avg surface temp
       temp = temp0   ! for starters
       P0 = 1013.0D0
       HREF = 7000.0D0
       TEMP1 = 273.2D0

       PRESS = P0 * EXP(-H/HREF)


       IF(H .LE. 11000.) THEN
       TEMP = TEMP0 - (LVALUE * H)
       XVALUE = 25.22 * ((TEMP - TEMP1)/TEMP)
     1 -     5.31 * DLOG( TEMP/TEMP1)
       EHUMID = 6.105 * EXP(XVALUE)
       ENDIF

       IF(H .GT. 11000.  .AND.  H .LT. 20000.) THEN
C      TEMP STAYS CONSTANT IN TROPOPAUSE AS DOES EHUMID

!      write(*,*) "got to here 1 ", h, hp1


!      write(*,*) "got to here 1a ", temp,temp1,theta


       XVALUE = 25.22 * ((TEMP - TEMP1)/TEMP)
     1 -     5.31 * DLOG( TEMP/TEMP1)
       ENDIF


       IF(H .GE. 20000.) THEN
C      EHUMID STAYS CONSTANT HERE AND TEMP GOES UP LINEARLY
       TEMP = TEMP + 0.002D0 *(H - 19999)
       XVALUE = 25.22 * ((TEMP - TEMP1)/TEMP)
     1 -     5.31 * DLOG( TEMP/TEMP1)
       ENDIF
```

```
      NFRACT = 77.6 * PRESS/TEMP +
     1        3.73 * (10.0**5) * (EHUMID/TEMP**2)


      N = (10.0**(-6) * NFRACT) + 1.0D0



     .DPHI = DL/(RE + H)
      DH = (RE + H)*DTAN(THETA)*DPHI
      HP1 = H + DH
      PRESS = P0 * EXP(-HP1/HREF)

      IF(H .LE. 11000.) THEN
      TEMP = TEMP0 - (LVALUE * HP1)
      XVALUE = 25.22 * ((TEMP - TEMP1)/TEMP)
     1 -      5.31 * DLOG( TEMP/TEMP1)
      EHUMID = 6.105 * EXP(XVALUE)
      ENDIF


      IF(H .GT. 11000.  .AND.  H .LT. 20000.) THEN
C     TEMP STAYS CONSTANT IN TROPOPAUSE AS DOES EHUMID
      XVALUE = 25.22 * ((TEMP - TEMP1)/TEMP)
     1 -      5.31 * DLOG( TEMP/TEMP1)
      ENDIF


      IF(H .GE. 20000.) THEN
      TEMP = TEMP0 + 0.002D0 *(HP1 - 19999)
      XVALUE = 25.22 * ((TEMP - TEMP1)/TEMP)
     1 -      5.31 * DLOG( TEMP/TEMP1)
      ENDIF



      NFRACT = (77.6 * PRESS/TEMP) +
     1        3.73 * (10.0**5) * (EHUMID/TEMP**2)
      NP1 = (10.0**(-6) * NFRACT) + 1.0D0

 55   CONTINUE
C     WRITE(23,*)H,N
!     write(*,*) "got to here 2", h, hp1,theta

      DNDH= (NP1 - N)/(HP1 - H)
      H = H - DH

      DTHETA = (1.0 + (DNDH/N)*(RE + H))*DPHI
      DS = ((RE + H)*DPHI)/DCOS(THETA)
      DSE = N*DS

      H = HP1
      THETA = THETA + DTHETA
      PHI = PHI + DPHI
      S = S + DS
      SE = SE + DSE
      IF(PHI.LT.PHI0)GO TO 800
c     WRITE(*,914)
c     1       H,THETA/RAD,PHI/RAD,PHI0/RAD
c914  FORMAT(1X,'H ',F12.4,' THETA ',F12.4,' PHI ',E12.4,
```

```
C      1 ' PHIO ',E12.4)
       IF(I.EQ.1) HA = H
       IF(I.EQ.2)THEN
          HB = H
          DHDTH = (HB - HA)/EPSLN1
          EPSLN2 =(H2 - HA)/DHDTH
       ENDIF

 100     CONTINUE
C      REWIND(23)
        THETA0 = TCOM + EPSLN2
c        THETA0 = 0.0
C*****************************************************************
C
C       OUTPUT S0,SE,THETA0,TCOM,LE,LC
C
C*****************************************************************

       ALFA = 0.5*BTRUE*RE
       C = ATRUE*RE*DEXP(-2.0*ALFA*H1/RE)
       FAC1 = 0.5*DSQRT(PI/ALFA)*DEXP(-ALFA*THETA0*THETA0)
       FAC2 = DSQRT(ALFA)*(PHI0 + THETA0)
       FAC3 = DSQRT(ALFA)*THETA0
       LE = C*FAC1*(ERF2(FAC2) - ERF2(FAC3))

       LC = (SE*(BTRUE*LE)**2)/24.

       RETURN
       END   !END TWGEN




       SUBROUTINE REFLTR(PHI,HX,PX,Q,R,X,Y,YEXP)
       IMPLICIT NONE
C*****************************************************************
C
C      REFLTR: GENERALIZED KALMAN FILTER EXECUTIVE ROUTINE
C      ADAPTED FROM FILTR9 BY RRD for new RE model which has a 3 element state
vector
C
C*****************************************************************

       REAL*8 PHI(3,3),HX(3,3),PX(3,3),Q(3,3),R(3,3),X(3),Y(3),
     1         YEXP(3)

       REAL*8 G(3,3),DUM(3,3),DUM1(3,3),DENOM(3,3),DUM2(3,3),DUM3(3,3)
       REAL*8 DUMX(3,3),M(3,3),D
       INTEGER*4 L2(3),NW(3),I,J
       LOGICAL FLAG


C-------------------------------------------------------------------------




C*****************************************************************
C
C      COMPUTE KALMAN GAIN MATRIX  G(3,3)
```

```fortran
C
C*********************************************************************
C assume transition matrix PHI = I
      PX(1,1) = PX(1,1) + Q(1,1)
      PX(2,2) = PX(2,2) + Q(2,2)
      PX(3,3) = PX(3,3) + Q(3,3)
C

      CALL DTMAML(PX,HX,DUM,3,3,3,3,0,1)
      CALL DTMAML(HX,DUM,DENOM,3,3,3,3,0,0)
      DO 202 I = 1,3
      DO 202 J = 1,3
      DUM1(I,J) = DENOM(I,J) + R(I,J)
 202     CONTINUE
      CALL DMINV1(DUM1,3,D)
      CALL DTMAML(DUM,DUM1,G,3,3,3,3,0,0)


C*********************************************************************
C
C        APPLY KALMAN STATE CORRECTION
C
C*********************************************************************

      X(1) = X(1) + G(1,1)*(Y(1) - YEXP(1))
      X(2) = X(2) + G(2,1)*(Y(1) - YEXP(1))
      X(3) = X(3) + G(3,1)*(Y(1) - YEXP(1))

C*********************************************************************
C
C        COVARIANCE MATRIX PX OBERVATION UPDATE
C
C*********************************************************************

      CALL DTMAML(G,HX,DUM2,3,3,3,3,0,0)
      DO 20 I = 1,3
      DO 20 J = 1,3
      DUM2(I,J) = -DUM2(I,J)
  20     CONTINUE
      DO 25 I = 1,3
      DUM2(I,I) = 1.0 + DUM2(I,I)
  25     CONTINUE
      CALL DTMAML(DUM2,PX,DUM3,3,3,3,3,0,0)
      DO 30 I = 1,3
      DO 30 J = 1,3
      PX(I,J) = DUM3(I,J)
  30     CONTINUE


      RETURN
      END   ! END REFLTR


      FUNCTION ERF2(Z)

C*********************************************************************
C
```

```
C        SERIES SOLUTION FOR ERROR FUNCTION ERF
C
C*******************************************************************

        REAL*8 ERF2,Y,Z,P,T,EY,A1,A2,A3,D

        Y = Z
        IF(Y.LT.0.)Y = -Y
        P = 0.47047
        T = 1.0/(1. + P*Y)
        EY = EXP(-Y*Y)
        A1 = 0.3480242
        A2 = -0.0958798
        A3 = 0.7478556
        D = 1.0 - EY*(A1*T + A2*T*T + A3*T*T*T)
        IF(Z.GE.0.0)ERF2 = D
        IF(Z.LT.0.0)ERF2 = -D
        RETURN
        END


C*******************************************************************
C      SUBROUTINE DTMAML( A, B, C, IRA, ICA, IRB, ICB, IFTRA, IFTRB )
C
C      This subroutine multiplies two double precision matrices (or their
transposes)
C      of arbitrary size. Matrices A,B,C must be physically distinct.
C      Output is C per the following table
C
C      IRA, ICA = Number of Rows/Cols of A
C      IRB, IRC  = Number of rows of B,C
C      IFTRA    IFTRB    IOP    C
C        0        0       1     A* B
C        1        0       2     AT * B
C        0        1       3     A* BT
C        1        1       4     AT * BT
C
C*******************************************************************


        SUBROUTINE DTMAML( A, B, C, IRA, ICA, IRB, ICB, IFTRA, IFTRB )

        IMPLICIT NONE
C
C      ! Input argument declarations.
        REAL*8      A(*), B(*)
        INTEGER*4   IRA, ICA, IRB, ICB
        INTEGER*4   IFTRA
        INTEGER*4   IFTRB

C      ! Output argument declarations.
        REAL*8      C(*)

C      ! Local declarations.
        REAL*8      CSUM
        INTEGER*4   NRA, NCA, NRB, NCB
        INTEGER*4   NA, NB, NC
        INTEGER*4   NXA, NXB, NYB
```

Page 191

```
      INTEGER*4  I, J, K
      INTEGER*4  IOP,IAA,IA,KAA,KAO,JBB,JBO,KBB,KBO,JCC,JCO,KC,JB,KA,
     $           KB,JC

C     ! Get the location (address) of C and see if its the same as that of A or
B.
C       IF ( %LOC(C) .EQ. %LOC(A) .OR. %LOC(C) .EQ. %LOC(B) ) THEN
C         WRITE( *, '('' Error in MAT_MULT - C is A or B'')' )
C         RETURN
C       END IF

C     ! Determine option.
      NRA = IRA
      NCA = ICA
      NRB = IRB
      NCB = ICB
      IOP = IFTRA + IFTRB + IFTRB + 1
      GO TO ( 2, 5, 8, 11), IOP

C     ! Conventional multiply  A * B.
 2    CONTINUE

      IAA = 1
      IA  = 0
      KAA = NRA
      KAO = -NRA
      JBB = NCA
      JBO = -NCA
      KBB = 1
      KBO = 0
      JCC = NRA
      JCO = -NRA
      NXA = NRA
      NXB = NCB
      NYB = NCA
      GO   TO 15

C     ! Transpose A only.
 5    CONTINUE

      IAA = NRA
      IA  = -NRA
      KAA = 1
      KAO = 0
      JBB = NRB
      JBO = -NRB
      KBB = 1
      KBO = 0
      JCC = NCA
      JCO = -NCA
      NXA = NCA
      NXB = NCB
      NYB = NRA
      GO   TO 15

C     ! Transpose B only.
 8    CONTINUE
```

Page 192

```fortran
        IAA = 1
        IA  = 0
        KAA = NRA
        KAO = -NRA
        JBB = 1
        JBO = 0
        KBB = NRB
        KBO = -NRB
        JCC = NRA
        JCO = -NRA
        NXA = NRA
        NXB = NRB
        NYB = NCA
        GO   TO 15

C       ! Transpose A and B.
   11   CONTINUE

        IAA = NRA
        IA  = -NRA
        KAA = 1
        KAO = 0
        JBB = 1
        JBO = 0
        KBB = NRB
        KBO = -NRB
        JCC = NCA
        JCO = -NCA
        NXA = NCA
        NXB = NRB
        NYB = NRA

C       ! Multiply the matrices.
   15   CONTINUE

        DO I = 1, NXA
          IA = IA + IAA
          JC = JCO
          JB = JBO
          DO J = 1, NXB
              JC = JC + JCC
              JB = JB + JBB
              KA = KAO
              KB = KBO
              CSUM = 0.
              DO K = 1, NYB
                  KA = KA + KAA
                  KB = KB + KBB
                  NA = KA + IA
                  NB = KB + JB
                  CSUM = CSUM + A(NA) * B(NB)
              END DO
              NC = JC + I
              C(NC) = CSUM
          END DO
        END DO
```

Page 193

```
      RETURN
      END


C*********************************************************************
C   SUBROUTINE DMINV1(A,N,D)
C
C      This routine inverts a matrix in place.
C
C      A is input NxN matrix to be inverted in place, i.e., output A = inverse of
input A
C
C      Limited to 20x20 matrices - to expand change L and M working vectors.
C*********************************************************************

      SUBROUTINE DMINV1(A,N,D)

      IMPLICIT NONE
C     ! Argument declarations.
      REAL*8 A(*)
      REAL*8 D
      INTEGER*4 N

C     ! Local declarations.
      REAL*8      BIGA, HOLD
      INTEGER*4   L(20),M(20),NK,KK,K,J,KJ,IJ,IZ,I,KI,JI,JP,JK,JR,JQ,IK
      LOGICAL     FLAG

C     ! Search for largest element.
      FLAG = .FALSE.
      D = 1.0
      NK = -N
      DO 80  K = 1, N

         NK = NK + N
         L(K) = K
         M(K) = K
         KK = NK + K
         BIGA = A(KK)
         DO 20 J = K, N
            IZ = N * (J-1)
            DO 20 I = K, N
              IJ = IZ + I
  10              IF ( ABS(BIGA) - ABS(A(IJ)) ) 15,20,20
  15              BIGA = A(IJ)
            L(K) = I
            M(K) = J
  20     CONTINUE

C        ! Interchange rows.
         J = L(K)
         IF ( J-K ) 35,35,25
  25     KI = K - N
         DO  I = 1, N
            KI = KI + N
            HOLD = -A(KI)
```

Page 194

```
            JI = KI - K + J
            A(KI) = A(JI)
            A(JI) = HOLD
        END DO

C          ! Interchange columns
   35      I = M(K)
        IF ( I - K ) 45,45,38
   38     JP = N * (I-1)
        DO J = 1, N
            JK = NK + J
            JI = JP + J
            HOLD = -A(JK)
            A(JK) = A(JI)
            A(JI) = HOLD
        END DO
C Divide column by minus pivot (value of pivot element is contained in biga)
   45      IF(BIGA) 48,46,48
   46      FLAG = .TRUE.
        WRITE (6,200)
  200      FORMAT(1H ,'THE MATRIX IS SINGULAR')
        RETURN
   48   DO 55 I = 1, N
            IF ( I - K ) 50,55,50
   50       IK = NK + I
            A(IK) = A(IK) / (-BIGA)
   55   CONTINUE

C          ! Reduce matrix
        DO 65 I = 1, N
            IK = NK + I
            IJ = I - N
            DO 65 J = 1, N
               IJ = IJ + N
               IF ( I - K ) 60,65,60
   60            IF ( J - K ) 62,65,62
   62          KJ = IJ - I + K
               A(IJ) = A(IK) * A(KJ) + A(IJ)
   65   CONTINUE
C         ! Divide row by pivot
        KJ = K - N
        DO 75 J = 1, N
            KJ = KJ + N
            IF ( J-K ) 70,75,70
   70       A(KJ) = A(KJ) / BIGA
   75   CONTINUE

C          ! Product of pivots
        D = D * BIGA

C          ! Replace pivot by reciprocal
        A(KK) = 1.0D00 / BIGA

   80   CONTINUE

C      ! Final row and column interchange
      K = N
```

Page 195

```
100   K = K-1
      IF ( K ) 150,150,105
105   I = L(K)
      IF ( I-K ) 120,120,108
108   JQ = N * (K-1)
      JR = N * (I-1)
      DO 110 J = 1, N
        JK = JQ + J
        HOLD = A(JK)
        JI = JR + J
        A(JK) = -A(JI)
110   A(JI) =HOLD
120    J = M(K)
      IF ( J-K ) 100,100,125
125   KI = K - N
      DO 130 I = 1, N
       KI = KI + N
       HOLD = A(KI)
       JI = KI - K + J
       A(KI) = -A(JI)
130    A(JI) = HOLD
       GO TO 100

150   RETURN
      END
```

14 APPENDIX E Software Requirements Specification for Inclusion of the Atmospheric Filter Kalman Filter Routine into the Link-16 OCP

Software Requirements Specification

Improvement of LINK-16 Navigation via Real-Time Atmospheric Modeling

Version 1.0

Prepared by
BAE SYSTEMS

February, 2003

DRAFT

Revision History

Name	Date	Reason For Changes	Version

14.1 Scope

14.1.1   Identification

This Software Requirements Specification (SRS) contains the software requirements for the Atmospheric Filter, as incorporated into the BAE Link-16 Navigation Simulator (LNS).

Prepared for the Department of the Navy: ONR 313

14.1.2 System overview

Range measurement errors of Link-16 PPLI messages are the largest contributors to Link-16 navigation performance. Inadequate modeling of atmospheric refraction is responsible for most of this. Although Link-16 terminals currently include an approximate compensation for atmospheric refraction, this is a static, 2-parameter model dating to 1971 which only considers average errors.

The system is not capable of compensating for ambient temperature, pressure, humidity and altitude variations. The prime objective of this project is to develop, test and evaluate real-time atmospheric calibration algorithms capable of reducing long range errors to the order of a few meters.

This system exploits BAE experience in sensor registration to use a so-called atmospheric filter capable of estimating atmospheric refraction model parameters with the goal of reducing errors in TOA estimation by one order of magnitude or more at up to 300 km true range and to an acceptable level for longer true ranges (> 300 km), and thus improve the performance of the existing navigation algorithms.

With a highly accurate measurement of the time of arrival (TOA) of a Link 16 message, uncertainties in the speed of light reduce the accuracy of the range estimation to the signal emitter to lower levels than what it could be if the speed of light were as accurately characterized as is now possible. The difference between the standard value of the speed of light (the speed at which an electromagnetic signal travels) in a vacuum and that in the atmosphere is due to the refractivity the signal experiences. The Atmospheric Filter improves the Link-16 range error calculation by estimating atmospheric refractivity in real time via a Kalman Filter. The Atmospheric Filter comes in two models, the simpler flat earth (FE) atmospheric model and the more realistic Ray Model or Round Earth (RE) model, each suited to certain emitter-receiver geometries. The estimate is developed from processing Self Navigation and Link-16 PPLI data. A future version of this software processing Refinement Filter output results is envisioned.

14.1.3 Document Overview

The purpose of this SRS is to specify in detail the requirements for the development, performance, and qualification of the Atmospheric Filter for proper execution of the requirements specified in the Atmospheric Filter Statement of Work (for ONR-313).

The SRS shall specify the following:

    a. Applicable documents
    b. Development and design requirements
    c. Qualification documents
    d. Preparation for delivery
    e. Notes
    f. Appendices

14.1.4 References

TBS

14.2 Overall Description

14.2.1 Project Perspective

The Atmospheric Filter is an estimator mechanized as a simple Extended Kalman Filter (EKF). The filter accepts PPLI and Self Navigation data to refine the speed of light estimation by accurately determining the atmospheric refractivity between the host system and other L16 members, thereby enhancing current navigation in range error reduction capabilities.

The proposed improvement indicates that the navigation range estimation performance of L-16 using this atmospheric filter process will reduce common position errors of a mobile community to very low levels, on the order of 1 meter.

14.2.2 Project Functions

The project has been divided into two phases of the development and demonstration of the desired precision navigation algorithm. This document specifies the test software for Phase 1.

Phase 1 – Detailed Design Study and Development of the algorithms. The development of a full engineering test bed for the integrated operation of inertial and L-16 navigation will be provided via a PC-based simulation tool exercising the various variants of the precision navigation algorithm.

Phase 2 – Extension of the demonstrated algorithm into actual flight hardware exercised on validated Terminal ATP assets to simulated required flight dynamics.

14.2.3 Operating Environment

A laboratory environment in which is implemented a real time algorithm combing INS and Link-16 input capable of achieving community positioning accuracy of O(1) meter will be used as a testbed. The operation of an integrated INS/Link-16 system requires SELF INS data and Supporting L16 PPLI message traffic.

Resolution: Incorporate the AF algorithm into BAE's Link 16 Navigation Simulator and GPS model to satisfy the requirements.

14.2.4 Design and Implementation Constraints

14.2.4.1 User Documentation
Reports and Data are to be in accordance with preparation and submission requirements as stated in the Statement of Work Document. The type of documents required are listed below:

1. Project Status Reports
2. Performance Reports
3. Cost Reports
4. Contract Funds Reports

14.2.4.2 Assumptions and Dependencies

TBD

14.3 Requirements

14.3.1 AF External Interface Requirements

This section specifies external software interfaces for the Atmospheric Filter. The Atmospheric Filter will interface with the Link-16 Operational Computer Program. As incorporated into the LNS (Link 16 Navigation Simulator), the AF_PROC will be called as a subroutine by the simulated OCP. It shall be called once per processing interval (i.e., once per second), at which time the output from the previous processing cycle will be read by the calling routine in the simulated OCP. The OCP will then send the routine the current processing cycle's Self Nav and PPLI data to process.

Inputs from the Link 16 OCP:
Self Navigation Solution: SELF NAV LAT, LON, ALT, TC, N. Velocity, E. Velocity, Height rate.
PPLI data: PPLI LAT, LON, HGT, TOA, TOV
Control input.

Outputs to the Link 16 OCP:
AF_RESULT_MSG: AF Filter Solution Results Message.
STN_STATUS: AF STN Status Message for Host (what STNs are in what model).

For Input and Output details, see Table 1.

```
┌──────────────┐   PPLI  LAT, LON, HGT, TOA, TOV    ┌──────────────┐
│   Link -16   │ ─────────────────────────────────► │  Atmospheric │
│  Operational │                                    │    Filter    │
│   Computer   │   OWN NAV  LAT, LON, ALT, TOV,      │  Processing  │
│   Program    │   N. Velocity, E. Velocity, ALT rate│  (AF_PROC)  │
│              │ ─────────────────────────────────► │              │
│              │              Control               │              │
│              │ ─────────────────────────────────► │              │
└──────────────┘                                    └──────────────┘
       ▲                                                   │
       └───────────────────────────────────────────────────┘
            AF_RESULTS_MSG
            STN_STATUS
```
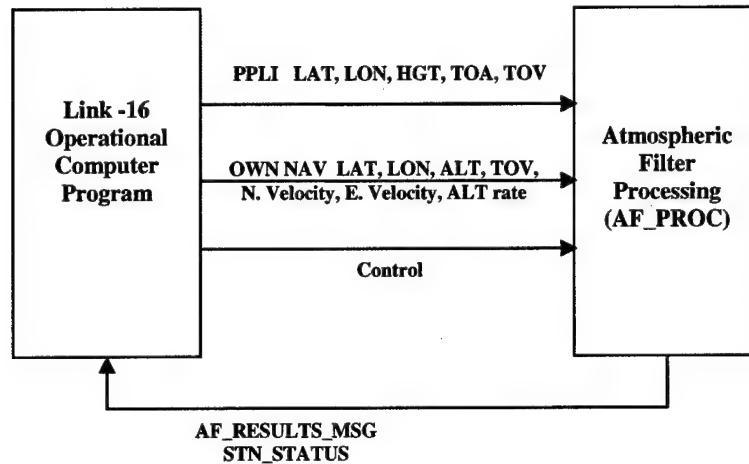
Figure 83 AF Top level system architecture (Level 0)

Table 10 ATMOSPHERIC_FILTER External Interfaces

Interface	Definition	Description	Reference	Input/Output
CONTROL_FLAG	START_AF_OPER STOP_AF_OPER	Reset/Initiate flag from OCP.		Input
NAV_ARRAY	LAT, LON, HGT, V_NORTH,V_EAST, V_UP, TC	Self Navigation solution, from L16 OCP.		Input
PPLI_ARRAY	LAT, LON, HGT, TOA, STN, TOV	PPLI message, from OCP.		Input
AF_RESULTS_MSG	Model number, List of STNs in Model, Refractivity coefficients a & b, and Quality Q	Indicates health of the Atmospheric Filter to the host. A full results message including, the observation processed, model used, time, refractivity coefficients and covariance matrix values.		Output
STN_STATUS (J_FILT,k)	Model number J_FILT = 1, ... ,6 and k = 1, N (J_FILT)	What STN is associated with what model.		Output

14.3.2   AF Architecture

The Atmospheric Processing is organized into six operational CSC's depicted below.
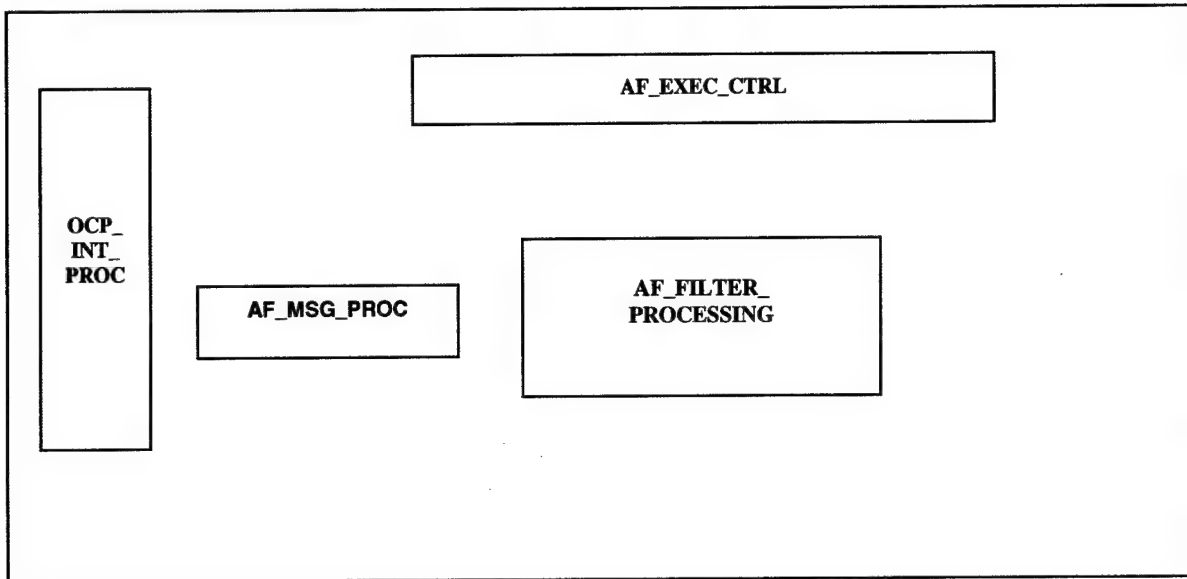
Figure 84 AF_PROC Level 1 operational functions

Table 11 ATMOSPHERIC_PROC CSC Descriptions

CSC	Description
AF_EXEC_CTRL	The functional group that controls the data flow within the Atmospheric Processing depending on data coming into the system. Screens data, monitors status, generates report messages. Writes Nav and PPLI data to tables for input to AF_FILTER_PROC.
AF_ OCP_INT_PROC	The L16 OCP interface from which all OCP data enters the system: navigation solution including SELF Nav position, quality, velocity and time, PPLI position, and quality. For incorporation into the LNS, this shall be a wrapper routine for a subroutine call by the OCP.
AF_ FILTER_PROC	Prepares and synchronizes all data in the system and packages them for processing by the Kalman Filter. Determines which KF model to use. Applies Kalman filtering to the data passed from the AF_EXEC_CTRL unit.
AF_MSG_PROC	The CSC which is called at the end of each time step of one second's duration. This block composes a Packed 4 message of data to be used to correct TOA, contained within the Link-16 Operational Computer Program, if the data is from a new unit to be processed, the message will instruct the OCP to use the original TOA update routine. It also sends a full results message to the Host every second.

14.3.2.1 AF_EXEC_CTRL

The Atmospheric Filter Executive Control (AF_EXEC_CTRL) serves as an orchestrator for the entire Atmospheric Filter Process. The filter is broken down into 2 states within the AF executive control. Depending on the health of data coming into the system and the control inputs from the host, the AF executive control determines the state of the filter and performs corresponding operations.
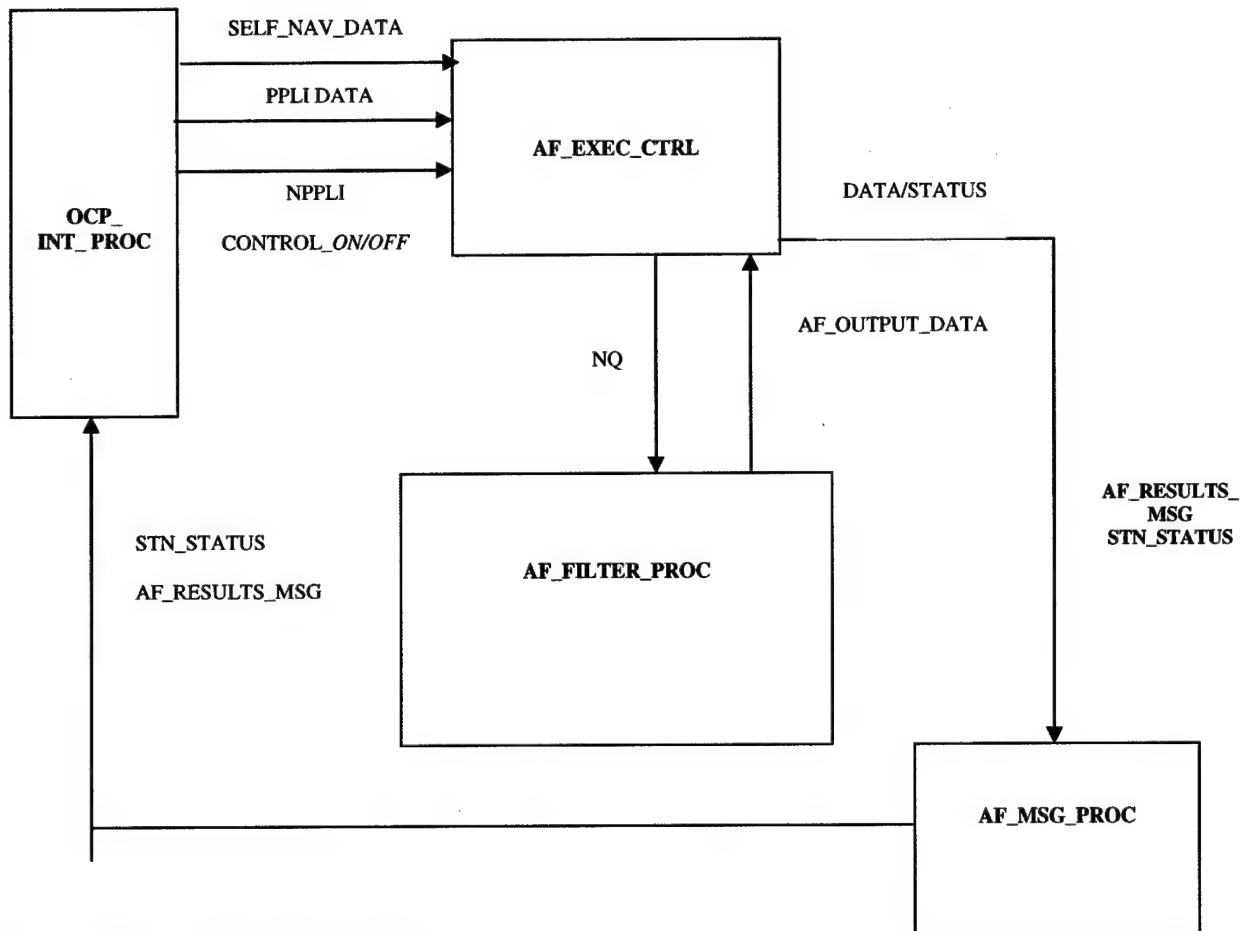
Figure 85 Level 1 AF_EXEC_CTRL

Inputs to AF_EXEC_CTRL:

From OCP_INTERFACE_PROCESSING:

1. NPPLI – Number of pending PPLI messages to process

2. PPLI_ARRAY - NPPLI Blocks of: PPLI data and quality, TOA and TOV

3. NAV_ARRAY - Valid SELF_NAV DATA and Quality; TC (= Time of Validity)

4. CONTROL_FLAG - AF On/Off Select

From AF_KALMAN_PROC:

1. STN_STATUS - 2D array

2. AF_RESULTS_MSG - 3D array

Purposes / Functional Description:

1. To create and maintain status tables of participating STN platforms and classify them as to the appropriate AF operational filter class,
2. To screen for quality and organize incoming self-navigation and received PPLI navigation data for later processing by AF_SOURCE_DATA_PROCESSING.
3. To control the operation of the various AF Kalman Filter algorithms according to the status and arrival of the corresponding PPLI navigation data.
4. To invoke the AF_RESULTS data package for transmission to L16 OCP for application to the speed of light correction.
5. To generate an AF_STATUS message for transmission to the Host which contains the AF state vector, covariances, and participating STNs for each of the six AF filter classes.
6. To Monitor for and respond to OCP provided On/Off control commands.
7. To Monitor health of AF processing.

AF_EXEC_CTRL Functional Block Breakdown

(a) CLASSIFY_PPLI_MSGS
Evaluates quality of PPLI data and organizes into AF filter classes depending on relation to Self Nav data.

(b) GEN_STATUS_REPORT
Prepares status report information in AFSV

Operational Rules for Application of AF Corrections to OCP Speed of Light:

(a) No AF speed of light correction for any STN shall be applied to the OCP until steady state operation of that STNs AF filter has been achieved. (Filter Q > 10)

(b) No processing of any of the six AF filters shall be permitted unless Own (Self) Navigation is valid, and a relative quality(Q_{PR}) level of at least 10 has been achieved. Once this level has been achieved, processing in that filter will continue while the quality remains above 8.

(c) No STN's data shall be accepted for AF processing unless it has achieved a relative quality level of at least 10.

(d) The observation variance, R, used for processing of any STN shall be a function of that STN's Q_{PR} and Own Q_{PR}.

(e) The speed of light correction for any STN not identified with a steady state AF filter shall use the existing (i.e., default) algorithm.

Indicator and Control Variables:

1) AF_STATUS_VECTOR: ASFV(J_FILT); J_FILT = 1, …, 6, where J_FILT = index of each of the six AF filters. The significance of the ASFV is described in the Table.

Table 12 AF_STATUS_VECTOR

AFSV(J_FILT)	Description
0	Filter currently *inactive*. That is, no candidate STN has yet been identified for that computation. This is the initial condition on startup.
1	Filter has received first appearance of any STN, but *has not yet been initialized.*
2	Filter initialized, but has *not yet reached steady state*. Q < 10
3	Filter in *steady state*. Q >= 10

2) STN_STATUS (J_FILT,k): List of specific STNs associated with filter J_FILT, where k = 1, N(J_FILT). Note that this table *must be provided to OCP* to identify which STNs are eligible for the speed of light correction, and with which parameter set.

Operational Note:

- 208 -

The AF routine will be called at the rate of 1 HZ. During that time, many PPLI messages may arrive at the OCP interface processing. Therefore, multiple AF filter updates are likely at that time step.

Pseudocode:

Start: Check for 1) Valid Self Nav data and 2) if there are any PPLIs to process this processing cycle. If 1) and 2) are true, then call CLASSIFY_PPLI_MSGS, after which, call AF_FILTER_PROC, then call AF_RESULTS_GEN, and then finally call GEN_STATUS_REPORT, then return.

Outputs:
STN_STATUS(J_FILT) @ 1HZ to OCP
KF State/Quality values for all six filter classes.

PPLI Data in appropriate KF model tables (to AF_SOURCE_DATA_PROC) in form of NQ array @ 1 HZ (to OCP)

Table 13 AF_EXEC_CTRL

CSC Inputs/Outputs	Input or Output	Description
START_AF_OPER	Input	Control Signal: Starts the AF processing. From AF_OCP_INT_PROC
STOP_AF_OPER	Input	Control Signal: Stops the AF processing. From AF_OCP_INT_PROC
NPPLI	Input	Number of PPLI to process this (1 HZ) cycle. From AF_OCP_INT_PROC
AFSV(J_FILT)	Output	AF_STATUS_VECTOR, AF State Vector Status values, 0,1, 2, 3 give state of filter number J_FILT, where J_FILT = 1, 2, 3, 4, 5, or 6. See Table 3 for details. To AF_FILTER_PROC
AF_RESULTS_MSG	Output	Output of the Atmospheric Filter to the OCP. A full results message including model used, time, refractivity coefficients and covariance matrix values or quality indications. To OCP
STN_STATUS	Output	What STN is associated with what model. To OCP
NQ(J_FILT,k,m)	Output	3D array of data for each of the six KF models. To AF_FILTER_PROC
NUPD	Output	Number of pending updates this cycle. To AF_FILTER_PROC

Figure 86 NQ (J_FILT,k,m) Table

(Output from AF_EXEC_CTRL, Input to AF_FILTER_PROC)

(Table shows J_FILT [th] filter layer, max value of k = number of observations to process per filter J_FILT)

m Index

→

k I n d e x	Observation, k, in model J_FILT k_{max} = N(J_FILT)	m =1 TOV (counts)	2 LAT (rad)	3 LO N (rad)	4 ALT (H) (feet)	5 Θ_0 (rad)	6 Meas. Quality Q_R (dimensio nless, 0- 15)
	k = 1						
	k = 2						
	k = 3						
	k = 4						
	k = 5						
	....						

E.g., NQ(1,4,2) = LAT of 4th observation in model 1.

AF_OCP_INTF_
PROC

AF_MSG_PROC

CLASSIFY_PPLI_MSGS

GEN_STATUS_REPORT

AF_FILTER_PROC
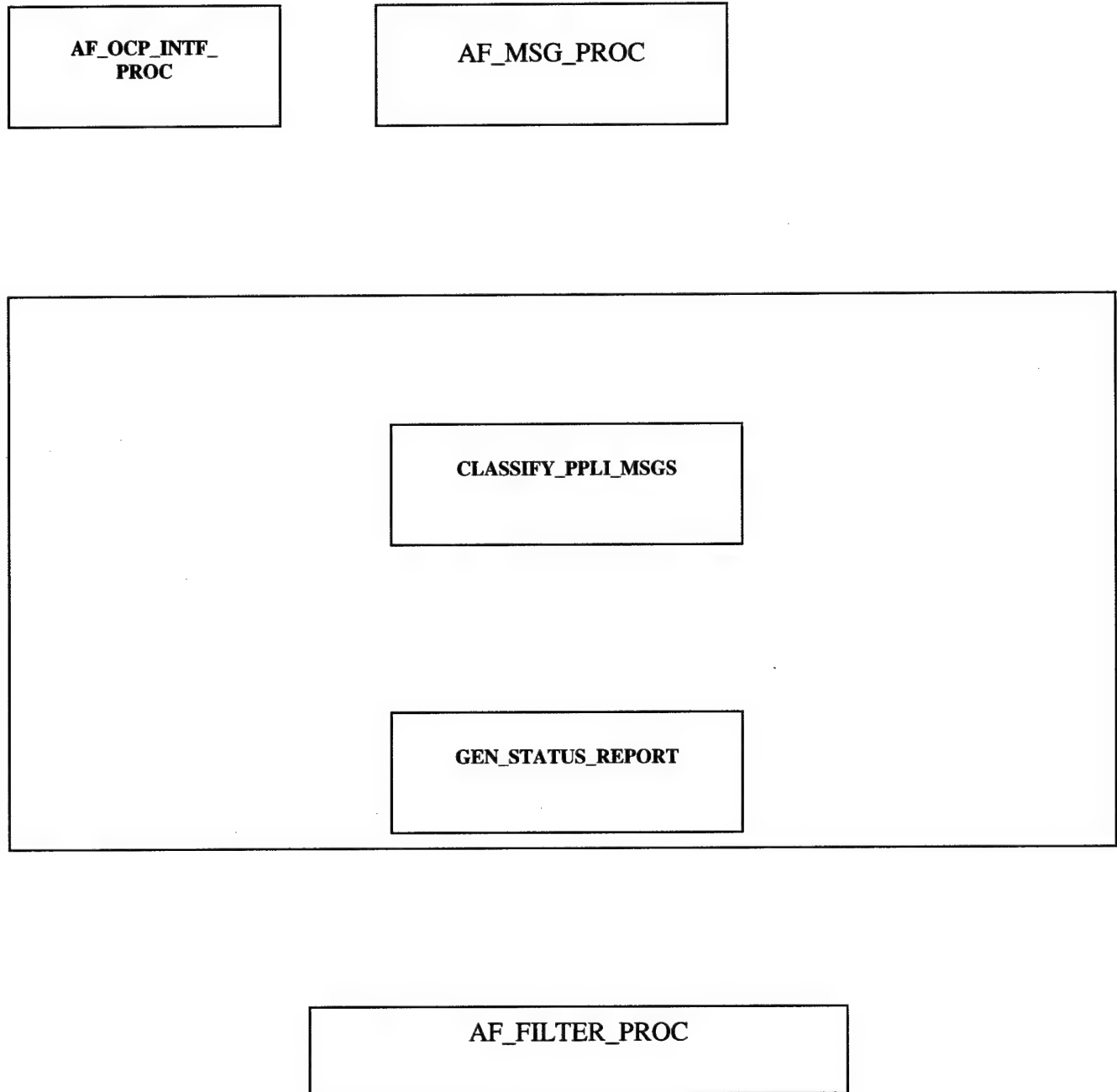
Figure 87 Level 2 AF_EXEC_CTRL Processing Operational Functions

STN_STATUS

AF_RESULTS_MSG
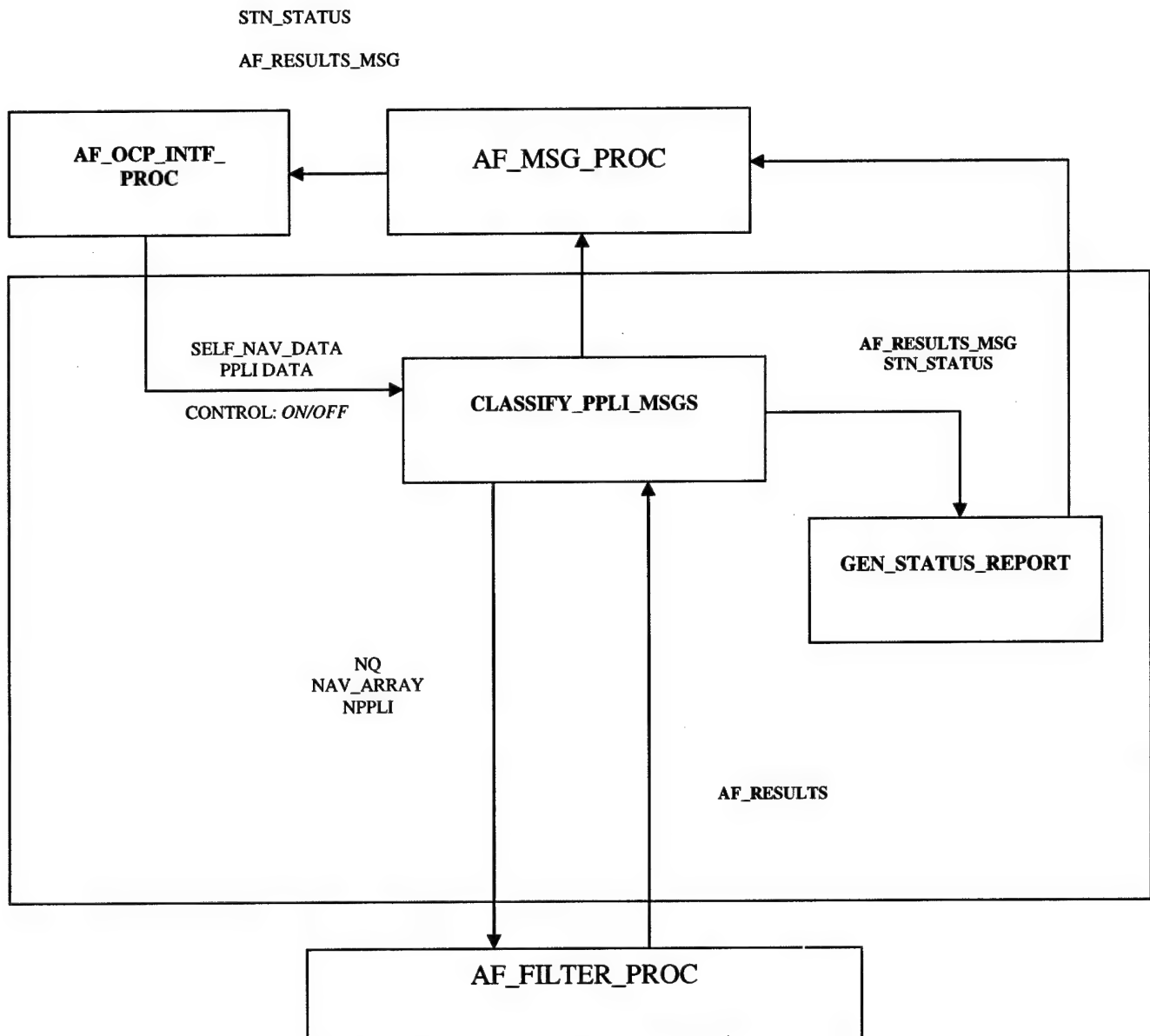
Figure 88 Level 2 AF_EXEC_CTRL Processing Functional Flows

START

CALL AF_OCP_INT_PROC

AF_OCP_INT_PROC is basically a wrapper for a subroutine call from OCP. It checks for Self Nav validity and counts the PPLIs available

SELF_NAV valid & $Q_{Self\ nav} > 10$?

NO → B

YES

N_PPLI > 0 ?

NO → B

YES

CALL CLASSIFY_PPLI_MSGS

UPDATE: AFSV(J_FILT), STN_STATUS(J_FILT,k)

Other Outputs:

NUPD - # of pending updates this cycle.

CURRENT_STN_STATUS(J_FILT,k).

For Each Candidate:

Read P_TOV (time of validity)

NQ
NAV_ARRAY
NPPLI

A   B

CALL AF_FILTER_PROC

CALL GEN_STATUS_REPORT

CALL AF_GEN_RESULTS_MSG

EXIT

A

Figure 89   AF_ EXEC_CTRL PROCESSING: LEVEL 2 Flowchart

14.3.2.2 CLASSIFY_PPLI_MSGS

Inputs:
NAV_ARRAY
PPLI_ARRAY

Purpose:

Classifies PPLI messages into AF Filter classes. Do this *before* extrapolating Nav data to PPLI TOV.

Pseudocode:

- Initialize NQ array

- Compute Approximate Included Angle PHI0 between Self & PPLI Platform

- Compute Approximate Range S0

- Compute Initial Estimate for THETA0

- Apply Classification Based on S0, THETA0 - Assign J_FILT to STN

- Update AFSV (AF Status Vector)

Output:
NPPLI
N(1), ... , N(6)
NQ

START

Zero N (J_FILT),
NQ (J_FILT,k,m)
Arrays

N (J_FILT) = No. of observations in current
Filter J_FILT queue
NQ (J_FILT,k,m) = Data Array

Loop for all l:
l = 1, NPPLI

Compute Approx.
Included Angle Φ_0
between Self & PPLI
Platform

$$\Phi_0 = \cos^{-1} [\cos \lambda_s * \cos \lambda_l * \cos(\Phi_s - \Phi_l) + \sin \lambda_s * \sin \lambda_1]$$

Compute
Approx. Range S_0

$$SPHI2 = \sin^2 (\varphi_0 / 2.)$$

$$S_0 = [(h_s - h_1) + 4. * (R_E + h_s) * (R_E + h_l) * SPHI2]^{1/2}$$

Compute
Initial Estimate for
Θ_0

$$\Theta_0 = \sin^{-1} ((h_l - h_s (S_0^2 / (2. * R_E))) / S_0)$$

Apply Classiifcation
Based on S_0, Θ_0
- Assign J_FILT

Update STN_STATUS(j,k)
N(J_FILT) = N(J_FILT) + 1

STN previously
assigned to
another J_FILT
?

YES → Delete Entry In STN_STATUS(J_FILT
*, k),
Where J_FILT * = previous index
assigned to STN_l

NO

AFSV(j) = 0
?

YES → AFSV(J_FILT) = 1

NO

Save TOV_l, λ_l, φ_l, H_l, θ_0 in NQ(J_FILT,k,m) array

End Loop

EXIT

Figure 90 CLASSIFY_PPLI_MESSAGES

- 215 -

14.3.2.2.1 Zero N (J_FILT), NQ (J_FILT,k,m) Arrays

Inputs:

Purpose:

Initializes arrays.

Pseudocode:

For J_FILT= 1, 6

N(J_FILT) = 0

      For k = 1, KMAX
           For m = 1, MMAX

           NQ (J_FILT,k,m) = 0

           END FOR
      END FOR
END FOR

14.3.2.2.2 Compute Approx. Included Angle Φ_0 between Self & PPLI Platform

Inputs:

Purpose:

Compute included angle PHI0 between platforms.

Calculate $\text{PHI0} = \cos^{-1}[\cos \lambda_{self} * \cos \lambda_{PPLI} * \cos(\Phi_{self} - \Phi_{PPLI}) + \sin \lambda_{self} * \sin \lambda_{PPLI}]$

Pseudocode:

PHI0 = DACOS(DCOS(S_LAT)*DCOS(P_LAT)*(DCOS(S_LON – P_LON)) + DSIN(S_LAT)*DSIN(P_LAT))

14.3.2.2.3 Compute Approx. Range S_0

Inputs:

<u>Purpose:</u>

Compute true range value S_0. Where

$$SPHI2 = (\sin(PHI0 / 2.))**2$$

$$S_0 = [(h_s - h_1) + 4. * (RE + S\_HGT) * (RE + P\_HGT) * SPHI2]^{1/2}$$

<u>Pseudocode:</u>

```
SPHI2 = DSIN(0.5*PHI0)**2
S0 = DSQRT((S_HGT - P_HGT)**2 + 4.0*(RE+ P_HGT)*(RE+ S_HGT)*SPHI2)
```

14.3.2.2.4   Compute Initial Estimate for Θ_0

<u>Inputs:</u>
S_0, heights.

<u>Purpose:</u>

Compute initial estimate for theta0. $THETA0 = \sin^{-1} ((h_1 - h_s (S_0^2 / (2. * R_E))) / S_0$

<u>Pseudocode:</u>

```
THETA0 = DASIN((P_HGT- S_HGT - (S0**2/(2.*RE)))/S0
```

14.3.2.2.5   Apply Classification Based on S0, THETA0 - Assign J_FILT

<u>Inputs:</u>

Purpose:

Assign a filter model to an STN, based on the following:

Model	J_FILT	THETA0 Values (degrees)	S0 (Range) Values (km)
FE1	1	> 70 or < -70	any
FE2	2	> 50 and < 70 or <-50and >-70	< 50
FE3	3	> -50 and < 50	< 50
RE1	4	> 50 and < 70	> 50
RE2	5	> -50 and < 50	> 50
RE3	6	> - 70 and < -50	> 50

Table 14 Filter Assignments, based on THETA0 and S0

Pseudocode:

```
Update STN_STATUS (J_FILT,k) array
N(J_FILT) = N(J_FILT) + 1
FOR J_FILT = 1, NUM_MODELS
        STN_STATUS(J_FILT,k) = STN
END FOR
```

14.3.2.2.6   Update AF Status Vector
Inputs:

AFSV(J_FILT)

Purpose:

Update AFSV(J_FILT) value.

Pseudocode:

```
If AFSV(J_FILT) = 0
Then
Set AFSV(J_FILT) = 1
ENDIF
```

Output:

AFSV(J_FILT)

14.3.2.2.7 GEN_STATUS_REPORT

Inputs:

PPLI_Data

Purpose:

Prepare AFSV(J_FILT) to be sent to OCP via AF_MESSAGE_PROC.

Pseudocode:

For each J_FILT:

Assign AFSV(J_FILT) = 0,1,2, or 3 as follows:

0	Filter currently *inactive*. Initial condition on startup. That is, no candidate STN has yet been identified for that computation.
1	Filter has received first appearance of any STN, but *has not yet been initialized.*
2	Filter initialized, but has *not yet reached steady state*. That is Q < 10
3	Filter in *steady state*. Q >= 10

<u>Output:</u>

AFSV vector

14.3.2.3 AF_FILTER_PROC

<u>Inputs:</u>

NAV_ARRAY:
S_LAT,S_LON,S_HGT,S_V_NORTH,S_V_EASTS_V_UP, S_TC (NAV Time of Validity)
NPPLI
N array, N(1), … , N(6)
NQ(J_FILT,k,m)

<u>Purpose:</u>

Run appropriate Kalman Filter: FE (if J_FILT = 1,2,3) or RE (if J_FILT = 4, 5,6). Evaluate quality of solution for a given J_FILT.

<u>Pseudocode:</u>

START

INPUT NQ array of
PPLI data,
NAV_ARRAY array
of Self NAV data

Begin Loop for J_FILT =
1, 6

N(J_FILT) =
NQ(J_FILT,k,1)

C   NO

$N(J\_FILT) > 0$?

YES

Begin Loop for I = 1,
N(J_FILT)

P_TOV = NQ(J_FILT,l,1)

Compute Extrapolation Time

$\Delta t = P\_TOV - S\_TC$
Where TC =
NAV_ARRAY(7)

Extrapolate Self Position to
P_TOV

$\lambda_{sE} = \lambda_s + \Delta t * v_{Ns} / R_E$

$\varphi_{sE} = \varphi_s + \Delta t * v_{Es} / (R_E * \cos \lambda_s)$

Note:
$\lambda_l = NQ(J\_FILT,l,2)$
$\varphi_l = NQ(J\_FILT,l,3)$

Compute Included angle Φ_0
between Self and PPLI STN

$\Phi_0 = \cos^{-1} [\cos \lambda_{sE} * \cos \lambda_l * \cos(\Phi_{sE} - \Phi_l) + \sin \lambda_{sE} * \sin \lambda_l]$

$SPHI2 = \sin^2 (\Phi_0 / 2.)$

Compute Predicted Range S_0

$S_0 = [(h_{sE} - h_l) + 4. * (R_E + h_{sE}) * (R_E + h_l) *$

- 222 -

Invoke Ray Model to
Compute Approximate True
Range

TMEAS =
Float[INT(S_0/KC)}
KC = 3.74705725

A
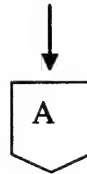
Figure 91 AF_FILTER_PROC Level 2 Flowchart- Part 1

$Y = TM$

FE

$J\_FILT = 1.2.3$

$J\_FILT = 4.5.6$

RE

MODEL TYPE
RE or FE?

$YEXP_{FE} = (S_0/KC)/(1.0 - X_F(1) + 0.5*X_F(1)*X_F(2)*(h_{sE}+h_I))$

X_F = State vector element from FE filter J_FILT

Compute $YEXP_{FE}$

Compute H_F matrix

$YEXP_{RE} = FLOAT(INT((L_{EA} + L_{CA} + S_0))/KC)$

Where:
$L_{EA} = K*X_R(1)* exp(-2.*X_R(2)*(\theta_0^2 + H_{sE}/R_E))*PHI0$

$B = 2.0*X_R(2)/RE$
$LC_A = (S0*(B*LEA**2)/24.0$

X_R = State vector element from RE filter j

Compute $YEXP_{RE}$

Compute H_R matrix

AFSV(J_FILT) = 1 ?

Initialize filter J_FILT, set AFSV(J_FI

Compute Observation

R elements are a function of Q_{RR}, Q_{RS}

Execute Kalman Update for filter

End Loop for i

C

End Loop for J_FILT

Exit

- 224 -

Figure 92   AF_FILTER_PROC Level2 Flowchart-Part2

14.3.2.3.1 Initialize_Filter

14.3.2.3.1.1 Initialize_Filter_FE

<u>Purpose:</u> Initialize FE Filter.

<u>Code:</u>

```
C*********************************************************************
C
C       GENERATE TRANSITION MATRIX PHI
C
C*********************************************************************

      PHI(1,1) = 1.0
      PHI(2,2) = 1.0
      PHI(2,1) = 0.0
      PHI(1,2) = 0.0


C*********************************************************************
C
C       INITIALIZE FLAT EARTH FILTER STATE VECTOR AND COVARIANCE MATRIX
C
C*********************************************************************

      DO 20 I = 1,2
      DO 20 J = 1,2
      PF(I,J) = 0.0D0
 20      CONTINUE
      PF(1,1) = SGA0**2
      PF(2,2) = SGB0**2

      XF(1) = A0
      XF(2) = B0

      YF(1)= 0.0
      YF(2)= 0.0


C*********************************************************************
C
C       INITIALIZE RATE LIMITING MATRIX FOR FLAT EARTH FILTER
C
C*********************************************************************

      DO 21 I = 1,2
      DO 21 J = 1,2
      QF(I,J) = 0.0
 21      CONTINUE

      QF(1,1) = 1.0E-5
      QF(2,2) = 1.0E-5
C*********************************************************************
C
C       INITIALIZE FLAT EARTH OBSERVATION VARIANCE MATRIX
```

```
C
C*********************************************************************

      DO 22 I = 1,2
      DO 22 J = 1,2
      RF(I,J) = 0.0
 22      CONTINUE

      RF(1,1) = 1.0E+0
      RF(2,2) = 1.0E+0
```

14.3.2.3.1.2 Initialize_Filter_RE

Purpose: Initialize RE Filter.

Code:
```
C*********************************************************************
C
C       INITIALIZE RAY MODEL KALMAN FILTER STATE, COVARIANCE & MEAS
C
C*********************************************************************

      DO 30 I = 1,FILT_DIM
      DO 30 J = 1,FILT_DIM
      PR(I,J) = 0.0D0
 30      CONTINUE

      PR(1,1) = SGGM0**2
      PR(2,2) = SGALF0**2
      PR(3,3) = SGK0**2
      PR(1,2) = 0.0
      PR(2,1) = 0.0
      XR(1) =  GAM0
      XR(2) =  ALF0
      XR(3) =  K

      YR(1)= 0.0
      YR(2)= 0.0
      YR(3)= 0.0


C*********************************************************************
C
C       INITIALIZE RATE LIMITING MATRIX FOR RAY FILTER
C
C*********************************************************************

      DO 31 I = 1,FILT_DIM
      DO 31 J = 1,FILT_DIM
      QR(I,J) = 0.0D0
 31      CONTINUE

      QR(1,1) = 1.0E+0
      QR(2,2) = 1.0E+0
```

```
      QR(FILT_DIM,FILT_DIM) = 1.0E+0


C*******************************************************************
C
C        INITIALIZE RAY MODEL OBSERVATION VARIANCE
C
C*******************************************************************

      DO 32 I = 1,FILT_DIM
      DO 32 J = 1,FILT_DIM
      RR(I,J) = 0.0
 32      CONTINUE

      RR(1,1) = 1.0E+0
      RR(2,2) = 1.0E+0
      RR(1,2) = 0.0E+0
      RR(2,1) = 0.0E+0
      RR(3,3) = 1.0E+0
```

14.3.2.3.2   Extrapolate_Self_NAV_POS_TO_TOV

Purpose:

To extrapolate Self_NAV position from NAV_TC to PPLI_TOV.

PseudoCode:

Convert S_TC to counts, where S_TC = SELF_NAV(7)

$\Delta t = P\_TOV - S\_TC$

Extrapolated $\lambda_s = \lambda_s + \Delta t * v_{Ns} / R_E$

Extrapolated $\varphi_s = \varphi_s + \Delta t * v_{Es} / (R_E * \cos \lambda_s)$

Extrapolated $h_s = h_s + \Delta t * v_{UPs}$

14.3.2.3.2.1 Compute PHI0

Purpose:

Compute Φ_0 (in FORTRAN, PHI0) angle.

Code:
$$\Phi_0 = \cos^{-1} [\cos \lambda_{sE} * \cos \lambda_l * \cos(\Phi_{sE} - \Phi_l) + \sin \lambda_{sE} * \sin \lambda_1]$$

14.3.2.3.2.2 Compute S0

Purpose:

Compute S_0 value.

Code:
$$SPHI2 = \sin^2 (\Phi_0 / 2.)$$

$$S0 = [(h_{sE} - h_1) + 4. * (R_E + h_{sE}) * (R_E + h_l) * SPHI2]^{1/2}$$

14.3.2.3.2.3 Compute True Range

Purpose:

Compute True Range. Compute measured range to transmitting platform in 12.5 ns counts. Yf(1),yr(1) equal tmeas in 12.5 ns counts. True world model provides Se for measurement equation.

Code:
$$\text{TMEAS} = \text{Float}\ [\text{INT}(S_E/KC)\}$$
```
KC = 3.74705725

        TMEAS = FLOAT(INT(SE/KC))


        YF(1)= TMEAS
        YF(2)= 0.0
        YR(1)= TMEAS
        YR(2)= 0.0
        YR(3)= 0.0
```

14.3.2.3.2.4 Compute YEXP

Purpose:

Compute predicted observation array.

Code:

```
C*****************************************************************
C
C       COMPUTE PREDICTED OBSERVATION FOR FE FILTER
C
C*****************************************************************

        YFEXP(1) = (S0/KC)/(1.0 - XF(1) + 0.5*XF(1)*XF(2)*(H(I)+H(1)))
        YFEXP(2) = 0.0
```

14.3.2.3.2.5 Compute RE H Matrix

Purpose:

Compute observation matrix for RE model.

Code:
```
        SUBROUTINE HRMATR(HR,XR,H,TCOM,PHI0,KC,RE,SE)
       IMPLICIT NONE
C*****************************************************************
C
C       HRMATR GENERATES THE OBSERVATION MATRIX HR FOR THE
C       RAY MODEL (RE) KALMAN FILTER
C
C*****************************************************************

        REAL*8 HR(3,3),XR(3),H(4),TCOM,PHI0,KC,RE,SE
        REAL*8 FAC,FAC1,LEA,TMEAS,LETERM
        INTEGER*4 I,J

        TMEAS = SE/KC

        DO 10 I = 1,3
```

```
        DO 10 J = 1,3

        HR(I,J) = 0.0
10      CONTINUE

        FAC     = -2.* (TCOM*TCOM + (H(1)/RE))
        FAC1    = FAC * XR(2)
        LETERM  = XR(3) * XR(1) * DEXP(FAC1)* PHI0   ! = Le

        HR(1,1) = LETERM/(KC * XR(1))
        HR(1,2) = FAC * LETERM/KC
        HR(1,3) = LETERM/(KC * XR(3))

        RETURN
        END
```

14.3.2.3.2.6 Compute FE H Matrix

Purpose:

Compute FE H Matrix.

Code:

```
        SUBROUTINE HFMATR(HF,XF,H,S0,KC,JJ)
        IMPLICIT NONE
C*****************************************************************
C
C       HFMATR GENERATES THE OBSERVATION MATRIX HF FOR THE
C       FLAT EARTH (FE) KALMAN FILTER
C
C*****************************************************************

        REAL*8 HF(2,2),XF(2),H(4),S0,KC
        REAL*8 FAC1,FAC2
        INTEGER*4 JJ,I,J

        FAC1 = (1.0 - 0.5*XF(2)*(H(JJ) + H(1)))   !changed first + to -
        FAC2 = (1.0 - XF(1) + 0.5*XF(1)*XF(2)*(H(JJ)+H(1)))

        DO 10 I = 1,2
        DO 10 J = 1,2
        HF(I,J) = 0.0
10      CONTINUE


        HF(1,1) = (S0/KC)*FAC1/FAC2**2
        HF(1,2) = -(S0/2.*KC)*(XF(1)*(H(JJ)+H(1)))/FAC2**2

        RETURN
        END
```

- 231 -

14.3.2.3.2.7 Select R Matrix

<u>Purpose:</u>

There are two stored measurement uncertainty (R) matrices to choose from, one for best quality measurements of Quality >= 12 and one for acceptable quality measurements of 10 < Quality <12.

<u>PseudoCode:</u>

If Q .GE. 12
Then select R whose diagonal elements are of quality equivalent to 1 count

Else if Q .LE. 12 .AND. Q .GE. 10
Then select R whose diagonal elements are of quality equivalent to 3 counts

End if

14.3.2.3.2.8 Execute Kalman Update

<u>Purpose:</u>

Execute Kalman Update: COMPUTE KF GAIN, apply corrections, update covariance.

<u>Code:</u>

```
C******************************************************************
C
C     COMPUTE KALMAN GAIN MATRIX  G
C
C******************************************************************
FILT_DIM = 2 for FE or 3 for RE

   CALL DTMAML(PX,HX,DUM,FILT_DIM,FILT_DIM,FILT_DIM,FILT_DIM,0,1)
                        CALL
DTMAML(HX,DUM,DENOM,FILT_DIM,FILT_DIM,FILT_DIM,FILT_DIM,0,0)
                     DO 200 I = 1,FILT_DIM
                     DO 200 J = 1,FILT_DIM
                  DUM1(I,J) = DENOM(I,J) + R(I,J)
200   CONTINUE
                     CALL DMINV1(DUM1,FILT_DIM,D)
   CALL DTMAML(DUM,DUM1,G,FILT_DIM,FILT_DIM,FILT_DIM,FILT_DIM,0,0)


C******************************************************************
C
C     APPLY KALMAN STATE CORRECTION
C
C******************************************************************
FOR i = 1, FILT_DIM
      X(i) = X(i) + G(i,i)*(Y(i) - YEXP(i))
END FOR


C******************************************************************
**
C
C     COVARIANCE MATRIX PX OBERVATION UPDATE
C
C******************************************************************
**

      CALL
DTMAML(G,HX,DUMFILT_DIM,FILT_DIM,FILT_DIM,FILT_DIM,FILT_DIM,0,0)
      DO 20 I = 1,FILT_DIM
```

- 233 -

```
          DO 20 J = 1,FILT_DIM
          DUM2(I,J) = -DUM2(I,J)
   20   CONTINUE
          DO 25 I = 1,FILT_DIM
          DUM2(I,I) = 1.0 + DUM2(I,I)
   25   CONTINUE
          CALL
DTMAML(DUM2,PX,DUM3,FILT_DIM,FILT_DIM,FILT_DIM,FILT_DIM,0,0)
          DO 30 I = 1,FILT_DIM
          DO 30 J = 1,FILT_DIM
          PX(I,J) = DUM3(I,J)
   30   CONTINUE
```

14.3.2.3.2.9 Convert RE states to A_REFRAC & B_REFRAC

Purpose:

To convert the RE states to the refractivity coeffs a & b from gamma XRE(1) & alpha XRE(2).

Code:

A_REFRAC(J_FILT) = XR(1)/RE

B_REFRAC(J_FILT) = 2.0 * XR(2)/RE

Stored Constants:

RE = 6378137.0 [meters]	Radius of Earth

14.3.2.3.2.10 Utility Routines

DTMAML

Purpose:

This subroutine multiplies two double precision matrices (or their transposes) of arbitrary size. Matrices A,B,C must be physically distinct.
Output is C per the following table.

IRA, ICA = Number of Rows/Cols of A
IRB, IRC = Number of rows of B,C

IFTRA	IFTRB	IOP	C
0	0	1	A* B
1	0	2	AT * B
0	1	3	A* BT
1	1	4	AT * BT

Code:

```
SUBROUTINE DTMAML( A, B, C, IRA, ICA, IRB, ICB, IFTRA, IFTRB )
```

```
        IMPLICIT NONE
C
C       ! Input argument declarations.
        REAL*8     A(*), B(*)
        INTEGER*4  IRA, ICA, IRB, ICB
        INTEGER*4  IFTRA
        INTEGER*4  IFTRB

C       ! Output argument declarations.
        REAL*8     C(*)

C       ! Local declarations.
        REAL*8     CSUM
        INTEGER*4  NRA, NCA, NRB, NCB
        INTEGER*4  NA, NB, NC
        INTEGER*4  NXA, NXB, NYB
        INTEGER*4  I, J, K
        INTEGER*4  IOP,IAA,IA,KAA,KAO,JBB,JBO,KBB,KBO,JCC,JCO,KC,JB,KA,
     $             KB,JC

C    ! Get the location (address) of C and see if its the same as that
of A or B.
C       IF ( %LOC(C) .EQ. %LOC(A) .OR. %LOC(C) .EQ. %LOC(B) ) THEN
C          WRITE( *, '('' Error in MAT_MULT - C is A or B'')' )
C          RETURN
C       END IF

C       ! Determine option.
        NRA = IRA
        NCA = ICA
        NRB = IRB
        NCB = ICB
        IOP = IFTRA + IFTRB + IFTRB + 1
        GO TO ( 2, 5, 8, 11), IOP

C       ! Conventional multiply  A * B.
  2     CONTINUE

        IAA = 1
        IA  = 0
        KAA = NRA
        KAO = -NRA
        JBB = NCA
        JBO = -NCA
        KBB = 1
        KBO = 0
        JCC = NRA
        JCO = -NRA
        NXA = NRA
        NXB = NCB
        NYB = NCA
        GO  TO 15

C       ! Transpose A only.
  5     CONTINUE
```

```
        IAA = NRA
        IA  = -NRA
        KAA = 1
        KAO = 0
        JBB = NRB
        JBO = -NRB
        KBB = 1
        KBO = 0
        JCC = NCA
        JCO = -NCA
        NXA = NCA
        NXB = NCB
        NYB = NRA
        GO  TO 15

C       ! Transpose B only.
  8     CONTINUE

        IAA = 1
        IA  = 0
        KAA = NRA
        KAO = -NRA
        JBB = 1
        JBO = 0
        KBB = NRB
        KBO = -NRB
        JCC = NRA
        JCO = -NRA
        NXA = NRA
        NXB = NRB
        NYB = NCA
        GO  TO 15

C       ! Transpose A and B.
  11    CONTINUE

        IAA = NRA
        IA  = -NRA
        KAA = 1
        KAO = 0
        JBB = 1
        JBO = 0
        KBB = NRB
        KBO = -NRB
        JCC = NCA
        JCO = -NCA
        NXA = NCA
        NXB = NRB
        NYB = NRA

C       ! Multiply the matrices.
  15    CONTINUE

        DO I = 1, NXA
         IA = IA + IAA
         JC = JCO
         JB = JBO
```

```
        DO J = 1, NXB
            JC = JC + JCC
            JB = JB + JBB
            KA = KAO
            KB = KBO
            CSUM = 0.
            DO K = 1, NYB
                KA = KA + KAA
                KB = KB + KBB
                NA = KA + IA
                NB = KB + JB
                CSUM = CSUM + A(NA) * B(NB)
            END DO
            NC = JC + I
            C(NC) = CSUM
        END DO
        END DO

        RETURN
        END
```

DMINV1

<u>Purpose:</u>

This routine inverts a matrix in place.

A is input NxN matrix to be inverted in place, i.e., output A = inverse of input A. Limited to 20x20 matrices - to expand change L and M working vectors.

<u>Code:</u>

```
SUBROUTINE DMINV1(A,N,D)

        IMPLICIT NONE
C       ! Argument declarations.
        REAL*8 A(*)
        REAL*8 D
        INTEGER*4 N

C       ! Local declarations.
        REAL*8      BIGA, HOLD
        INTEGER*4   L(20),M(20),NK,KK,K,J,KJ,IJ,IZ,I,KI,JI,JP,JK,JR,JQ,IK
        LOGICAL     FLAG

C       ! Search for largest element.
        FLAG = .FALSE.
        D = 1.0
        NK = -N
        DO 80  K = 1, N

          NK = NK + N
          L(K) = K
          M(K) = K
          KK = NK + K
          BIGA = A(KK)
```

```
          DO 20 J = K, N
              IZ = N * (J-1)
              DO 20 I = K, N
                IJ = IZ + I
   10              IF ( ABS(BIGA) - ABS(A(IJ)) ) 15,20,20
   15              BIGA = A(IJ)
              L(K) = I
              M(K) = J
   20     CONTINUE

C         ! Interchange rows.
          J = L(K)
          IF ( J-K ) 35,35,25
   25      KI = K - N
          DO  I = 1, N
              KI = KI + N
              HOLD = -A(KI)
              JI = KI - K + J
              A(KI) = A(JI)
              A(JI) = HOLD
          END DO

C         ! Interchange columns
   35      I = M(K)
          IF ( I - K ) 45,45,38
   38      JP = N * (I-1)
          DO J = 1, N
              JK = NK + J
              JI = JP + J
              HOLD = -A(JK)
              A(JK) = A(JI)
              A(JI) = HOLD
          END DO
C Divide column by minus pivot (value of pivot element is contained in
biga)
   45      IF(BIGA) 48,46,48
   46      FLAG = .TRUE.
          WRITE (6,200)
  200      FORMAT(1H ,'THE MATRIX IS SINGULAR')
          RETURN
   48     DO 55 I = 1, N
              IF ( I - K ) 50,55,50
   50      IK = NK + I
              A(IK) = A(IK) / (-BIGA)
   55     CONTINUE

C         ! Reduce matrix
          DO 65 I = 1, N
              IK = NK + I
              IJ = I - N
              DO 65 J = 1, N
                 IJ = IJ + N
                 IF ( I - K ) 60,65,60
   60              IF ( J - K ) 62,65,62
   62           KJ = IJ - I + K
                 A(IJ) = A(IK) * A(KJ) + A(IJ)
   65     CONTINUE
```

```
C          ! Divide row by pivot
           KJ = K - N
           DO 75 J = 1, N
               KJ = KJ + N
               IF ( J-K ) 70,75,70
 70            A(KJ) = A(KJ) / BIGA
 75        CONTINUE

C          ! Product of pivots
           D = D * BIGA

C          ! Replace pivot by reciprocal
           A(KK) = 1.0D00 / BIGA

 80        CONTINUE

C       ! Final row and column interchange
       K = N
 100   K = K-1
       IF ( K ) 150,150,105
 105   I = L(K)
       IF ( I-K ) 120,120,108
 108   JQ = N * (K-1)
       JR = N * (I-1)
       DO 110 J = 1, N
         JK = JQ + J
         HOLD = A(JK)
         JI = JR + J
         A(JK) = -A(JI)
 110     A(JI) =HOLD
 120     J = M(K)
       IF ( J-K ) 100,100,125
 125   KI = K - N
       DO 130 I = 1, N
        KI = KI + N
        HOLD = A(KI)
        JI = KI - K + J
        A(KI) = -A(JI)
 130    A(JI) = HOLD
        GO TO 100

 150   RETURN
       END
```

ERF2

Purpose:

Series solution for error function erf.

Code:

```
FUNCTION ERF2(Z)

       REAL*8 ERF2,Y,Z,P,T,EY,A1,A2,A3,D
```

```
Y = Z
IF(Y.LT.0.)Y = -Y
P = 0.47047
T = 1.0/(1. + P*Y)
EY = EXP(-Y*Y)
A1 = 0.3480242
A2 = -0.0958798
A3 = 0.7478556
D = 1.0 - EY*(A1*T + A2*T*T + A3*T*T*T)
IF(Z.GE.0.0)ERF2 = D
IF(Z.LT.0.0)ERF2 = -D
RETURN
END
```

14.3.2.4 AF_OCP_INTF_PROC

Wrapper code providing an interface to a subroutine call from OCP subroutine to fetch this data as an argument list.

Purpose:

Gets data from OCP:

SELF_NAV array: S_LAT, S_LON, S_HGT, S_VE, S_VN, S_V_UP, S_TC, S_Q

PPLI_ARRAY : STN (Source ID Number), P_LAT, P_LON, P_ALT, P_TOA, P_TOV, P_TRANSMIT_QP

Gets control (on/off) flag, CONTROL_FLAG, from OCP.

PseudoCode:

In OCP:

Call AF_OCP_INTF_PROC

Fetch AF_RESULTS_MSG, NAV_ARRAY
Read NAV_ARRAY

If STN in it, then perform new speed of light correction for it
If STN not in it, then perform old speed of light correction for it

14.3.2.5 AF_MSG_PROC

Packages and sends messages

STN_STATUS and AF_RESULTS_MSG to OCP.

UNCLASSIFIED

Table 15 AF_RESULTS_MSG (Model and Filter ResultTable)

(Table sent to OCP @ once per processing cycle)

Model	STNs in Model	Refractivity Coefficients.		Quality Q
		A_REFRAC	B_REFRAC	
FE_1 (J_FILT = 1)	Set of {STNs in MODEL J_FILT}			
FE_2	…			
FE_3				
RE_1				
RE_2				
RE_3				

Packaged as:

J_FILT, STN_ARRAY(J_FILT), A_REFRAC(J_FILT), B_REFRAC(J_FILT), Q_SOLN

14.4 SRS Glossary

Table 16 Data Dictionary (Incomplete, blank fields are TBS)

DATA VARIABLE NAME (internal subroutine variables can be named identically to those in other subroutines, so refer to Home routine to avoid confusion)	DATA DESCRIPTION	Details + Range/Units	ORIGINATING or HOME ROUTINE FOR VARIABLE	Data Type
P_LAT	PPLI LAT	0 – PI Radians	Global	REAL*8
P_LON	PPLI LON	0 – TWO_PI Radians	Global	REAL*8
P_HGT	PPLI HGT	Feet	Global	REAL*8
P_TOA	PPLI TOA	counts	Global	REAL*8
P_STN	PPLI STN	0 to MAX_STNS	Global	Int*4
P_TOV	PPLI TOV	Counts	Global	REAL*8
P_TRANSMIT_QP	PPLI Quality	0-15 dimensionless	Global	INT*4
S_LAT	SELF NAV LAT	Radians	Global	REAL*8
S_LON	SELF NAV LON	Radians	Global	REAL*8
S_HGT	SELF NAV HGT	Feet	Global	REAL*8
S_V_NORTH	SELF NAV V_NORTH	Feet/sec	Global	REAL*8
S_V_EAST	SELF NAV V_EAST	Feet/sec	Global	REAL*8
S_V_UP	SELF NAV V_UP	Feet/sec	Global	REAL*8
S_TC	SELF NAV TC	Sec - Seconds from start of simulation (in test environment)	Global	REAL*8

S_Q	SELF NAV QUALITY	0-15	Global	INT*4
MAX_STNS		TBD - Maximum allowable number of STNs	Global	INT*4
PI	π	3.141592653589793	Global	REAL*8
TWO_PI	$2.0 * \pi$	6.283185307179586	Global	REAL*8
CONTROL_FLAG	Start/Stop flag for AF. Not needed if AF is a subroutine call?	Control Signal from OCP: Starts/Stops the AF processing	Global	2* BOOLEAN
NAV_ARRAY	S_LAT, S_LON, S_HGT, S_V_NORTH, S_V_EAST, S_V_UP, S_TC	Self Navigation solution, from L16 OCP	Global	7* REAL*8
PPLI_ARRAY	P_LAT, P_LON, P_HGT, P_TOA, P_STN, P_TOV, P_TRANSMIT_QP	PPLI message, from OCP	Global	6* REAL*8, INT*4
J_FILT	Filter index	J_FILT = 1,2,3 = FE, J_FILT = 4,5,6 = RE	Global	Int*4
STN_ARRAY(J_FILT)	An array of subarrays, there are J_FILT subarrays of N(J_FILT) elements each, constituting all the STNs for each model J_FILT		Global	J_FILT* N(J_FILT)* int*4's
START_AF_OPER	Start flag for AF. Not needed if AF is a subroutine call?	Control Signal from OCP: Starts the AF processing	Global	BOOLEAN
STOP_AF_OPER	Stop flag for AF. Not needed if AF is a subroutine call?	Control Signal from OCP: Stops the AF processing	Global	BOOLEAN
NPPLI		Input: Number of PPLI to process this (1 HZ) cycle	Global	Int*4
AFSV(J_FILT)		AF_STATUS_VECTOR, AF State Vector Status values, 0,1, 2, 3 give state of filter number J_FILT. See Table 3 for details.	Global	INT*4 array

AF_RESULTS_MSG	J_FILT, STN_ARRAY(J_FILT), A_REFRAC(J_FILT), B_REFRAC(J_FILT), quality,Q_SOLN, sent to L16 OCP	Indicates health of the Atmospheric Filter to the host. A full results message contains: Model number, List of STNs in Model, Refractivity coefficients a & b, and solution Quality Q_SOLN	Global	Int*4 NS(J_FILT) * Int*4 REAL*8 REAL*8 Int*4
STN_STATUS(J_FILT,k)	Model number J_FILT = 1, 6 and k = 1, N (J_FILT)	What STN is associated with what model.	Global	int*4's
N(J_FILT)		# of observations in model J_FILT	Global	6 * Int*4
NUPD(J_FILT)	No. pending updates this cycle per model J_FILT		Global	6*int*4
NQ(J_FILT,k,m)	For all models, array of Atmospheric Filter input parameters from AF_EXEC_CRTL	See Figure 4 for details	Global	Int*4, 6*Real*8
CURRENT_STN_STATUS(J_FILT,k)	Model number J_FILT = 1, 6 and k = 1, N (J_FILT)	What STN is associated with what model.	Global	Int*4's
Q_SOLN(J_FILT)	10-15 (smaller values are not allowed)	AF Solution quality	Global	Int*4
PHI0		Φ_0 in Figure 9		Real*8
S0		S_0 in Figure 9		Real*8
THETA0		θ_0 in Figure 9		Real*8
SPHI2		Intermediate variable for predicted range S0 calculation		
DASIN	Fortran library routine for arc sine		Global	Real*8
DACOS	Fortran library routine for arc cosine		Global	Real*8
DSIN	Fortran library routine for sine		Global	Real*8
DCOS	Fortran library routine for cosine		Global	Real*8
DSQRT	Fortran library routine for square root		Global	Real*8
RE	Radius of Earth	RE = 6378137.0 [meters]	Global	Real*8
PHI				Real*8
HX				Real*8

PX				Real*8
G				Real*8
Q				Real*8
R				Real*8
X				Real*8
Y				Real*8
YEXP				Real*8
DUM1				Real*8
DENOM				Real*8
DUM2				Real*8
DUM3				Real*8
				Real*8
DUMX				Real*8
M				Real*8
D				Real*8
L2				Real*8
NW				Real*8
I				Int*4
J				Int*4
DTMAML(A, B, C, IRA, ICA, IRB, ICB, IFTRA, IFTRB)	Utility routine to multiply two double precision matrices (or their transposes) Matrices A,B,C must be physically distinct. Output is C.		DTM AML	Real*8
A	Matrix to be multiplied		DTM AML	Real*8
B	Matrix to be multiplied		DTM AML	Real*8
C	Matrix A * B output from DTMAML		DTM AML	Real*8
IRA	Number of Rows of A		DTM AML	Int*4
ICA	Number of Cols of A		DTM AML	Int*4
IRB	Number of Rows of B		DTM AML	Int*4
ICB	Number of Cols of B		DTM AML	Int*4
IFTRA	Flag to multiply A transpose * B		DTM AML	Int*4
IFTRB	Flag to multiply A * B transpose		DTM AML	Int*4
DMINV1(A,N,D)	Utility routine to invert a matrix in place. A is input NxN matrix to be inverted in place, i.e., output		DMI NV1	Real*8

	A = inverse of input A. Limited to 20x20 matrices - to expand change L and M working vector dimensions.			
N	Working array dimension		DMI NV1	Real*8
A	Array to be inverted		DMI NV1	Real*8
L	20x20 working matrix		DMI NV1	Real*8
M	20x20 working matrix		DMI NV1	Real*8
ERF2	Utility routine to form series solution for error function ERF, Returns ERF2 value as result		ERF2	Real*8
Y	Internal variable		ERF2	Real*8
Z	Internal variable		ERF2	Real*8
P	Internal variable		ERF2	Real*8
T	Internal variable		ERF2	Real*8
EY	Internal variable		ERF2	Real*8
A1	Internal variable		ERF2	Real*8
A2	Internal variable		ERF2	Real*8
A3	Internal variable		ERF2	Real*8
D	Internal variable		ERF2	Real*8
				Real*8
				Real*8
FILT_DIM	Kalman Filter state vector dimension	2 for FE 3 for RE		Int*4
TMEAS				Real*8
SE	Approximate true range from Self to STN	m		Real*8
RAD	Number of degrees per Radian	57.2957795131 [dimensionless]	Global	REAL*8
CM	Speed of light in vacuo	2.99792458E8 [m/s]	Global	Real*8
KC	Speed of light in vacuo	3.747405725 [counts]	Global	Real*8
DELTA_T	Δt, time increment			Real*8
X(J_FILT)	KF state vector			Real*8
				Real*8

P(J_FILT)	KF Covariance array			Real*8
Q_KF(J_FILT)	KF Process Noise array			Real*8
				Real*8
SGGM0				Real*8
SGALF0				Real*8
SGK0				Real*8
GAM0				Real*8
ALF0				Real*8
Y(J_FILT)	Measurement in KF			Real*8
				Real*8
YEXP(J_FILT)	Extrapolated measurement in KF			Real*8
				Real*8
				Real*8
LEA				Real*8
LETERM				Real*8
FAC				Real*8
A_REFRAC(J_FILT)		Atmospheric refractivity parameter a for model J_FILT	Global	Real*8
B_REFRAC(J_FILT)		Atmospheric refractivity parameter b for model J_FILT	Global	Real*8
				Real*8

L16 Position Quality	Position Uncertainty (feet)	Error Bound (3 sigma) (feet)
15	< 50	< 150
14	71	213
13	100	300
12	141	423
11	200	600
10	282	846
9	400	1200
8	565	1695
7	800	2400
6	1130	3390
5	1600	4800
4	2260	6780
3	4520	13560
2	9040	27120
1	18018	54240
0	>18018	>54240

Table 17 L16 Position Quality

L16 Time Quality	Time standard deviation (nanosec)
15	< 50
14	71
13	100
12	141
11	200
10	282
9	400
8	565
7	800
6	1130
5	1600
4	2260
3	4520
2	9040
1	18018

0	>18018

Table 18 L16 Time Quality

14.4.1 Operational terms

The following is a summary of acronyms, abbreviations and terms used in this specification.

CANTPRO	Cannot Process
CCM	Communication Control Message
CI	Configuration Item
COR	Core CSCI
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DFD	Data Flow Diagram
DID	Data Item Description
DoD	Department of Defense
DTB	Data Transfer Blocks
DTI	Data Transfer Interrupt
ECR	Earth Centered Rotating
EOM	End of Message
EOS	End of Slot
I/O	Input/Output
Init	Initialization
IRS	Interface Requirement Specification
JTIDS	Joint Tactical Information Distribution System
kbps	Kilobits Per Second
L16	Link-16
LNS	Link 16 Navigation Simulation software system
MIDS	Multifunctional Information Distribution System
Hz	Hertz
ms	milliseconds
MSG	Message Processor CSCI
NAV	Navigation
nmi	Nautical Miles
NPS	Network Participation Status
OCP	Link-16 Operational Computer Program
OTTS	Original Transmit Time Slot
PEQ	Position Extrapolation Quality
PNC	Primary Navigation Controller

PPLI	Precise Participant Location and Identification
RELNAV	Relative Navigation
SAI	Slot Assignment Index
SDP	Software Development Plan
SOW	Statement of Work
SRS	Software Requirement Specification
STN	Source Track Number
STP	Software Test Plan
TADIL-J	Tactical Data Information Link - JTIDS
P_TOA	PPLI Time of Arrival
P_TOV	PPLI Time of Validity
S_TC	Self NAV Time of Validity
TSA	Time Slot Assignment
TSR	Time Slot Reallocation
U	Unclassified